EVALUATING THE EFFECT OF PROCESS VARIATION ON SILICON AND GRAPHENE NANO-RIBBON BASED CIRCUITS

BY

ARTEM A. ROGACHEV

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Adviser:

Associate Professor Deming Chen

ABSTRACT

With technology scaling, the variability of device parameters continues to increase. Both performance and power consumption are quite sensitive to process parameters (PP) such as length, width, doping, and oxide thickness. As a result it is critical to predict the effect of these process variations (PV) on the future manufactured die. Guard-banding is often used to safe-guard against these variations, but it is usually too pessimistic. An alternative is to perform Monte Carlo (MC) simulation. However, this can be very computationally expensive and impractical for large circuits with multiple design iterations. Statistical static timing analysis (SSTA) has been proposed to quickly estimate the performance of a circuit under process variations (PV). Even though this has been a well studied topic, to the best of the author's knowledge, no one has considered the impact of the statistical thermal profile during statistical analysis of the propagation delay. The first part of this work presents a SSTA tool which considers this interdependence and produces accurate timing estimation.

Besides traditional silicon transistors, graphene nano-ribbon (GNR) transistors are promising candidates for future scaled technologies. They offer high-mobility and mean free path and are very robust. Because these devices are very small, the impact of PVs is expected to be very large. Unlike CMOS, the evaluation of circuit-level impact of PVs on GNR field effect transistors is in very early stages. Regardless of whether SSTA, MC simulation, or guard banding is used, a compact, parameterizable, SPICE compatible model is necessary to enable any of those approaches. For this reason, the second part of this thesis focuses on the development of such a model, its verification, and application to circuit-level simulations.

To my parents and friends that made me the person I am today

ACKNOWLEDGMENTS

I would like to thank several people who have made significant contributions to this thesis and my personal development. First and foremost, I would like to thank my adviser, Professor Deming Chen. I joined his group as an undergraduate student and learned a great deal through my research projects. Professor Chen had confidence in me and gave me numerous opportunities to challenge myself. He provided support when I was stuck, but also gave me plenty of freedom to test and explore my own ideas. As a result, I experienced significant growth in my ability to solve problems, plan projects, work on teams, and to effectively communicate my ideas and findings. I would like to again thank him for the opportunities that I had in our group.

I would also like to thank my fellow graduate students in the lab for their help and support on the projects. Greg Lucas was first to introduce me to statistical timing analysis and provided the initial source code for my timing analysis project. Lu Wan, helped a great deal in setting up the experiments and providing other useful suggestions. I also had the pleasure of working with Christine Chen and Amit Sangai on the modeling of graphene nano-ribbon transistors. They performed most of the experiments and contributed greatly during discussions. I would also like to thank Giuseppe Iannaccone and Gianluca Fiori for their suggestions on modeling. Finally, I would like to thank everybody in our research group for their advice and support throughout my study.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Process Variation and Approaches for Modeling Their Effect

In deep submicron technologies it is becoming more and more challenging to fabricate devices with precise dimensions. Since the functionality of transistors is strongly dependent on its process parameters (PP), these imperfections in fabrication result in significant deviation of power and performance from their nominal values. For example, according to the International Roadmap for Semiconductors (ITRS) [1], the delay variability was estimated to be 49% in 2010 and will increase to 63% in 2015. It is clear that these variations cannot be ignored and must be accounted for. Typically the variation of effective length (L), oxide thickness (T_{ox}), width (W), and doping (N_D) are considered. In addition, the variations in each process parameter can have spatial correlations due to processes such as chemical-mechanical polishing.

The simplest solution is to use guard-banding. Essentially, timing analysis can be performed using the worst case device parameters. This would guarantee that the specifications are met, but may result in significant amount of overdesign, which would imply a power or an area penalty. An alternative is to target a certain performance yield. For example, the goal may be for 95% of the die to operate faster than 2GHz. One way to evaluate the performance yield of a circuit is to use Monte Carlo (MC) simulation. First the process parameters for each device are randomly chosen based on some probability function. Then the circuit is simulated to obtain the delay and power. Typically this process is repeated more than 10,000 times to obtain a histogram. Based on this histogram, the performance yield can be obtained. This method is accurate, but it can be very computationally expensive. If the user is working with a large circuit and going through multiple design iterations, the MC simulation may be too time-consuming.

Statistical static timing analysis (SSTA) has been proposed as an alternative to quickly estimate the performance yield of a circuit under process variations (PVs) [2], [3], [4]. In this case the delay of each gate is considered to be a random variable with a mean and a standard deviation. A standard timing graph approach is used where each gate corresponds to a node and each wire corresponds to an edge. In addition a source node is added, which connects to all of the inputs along with a sink node, which connects to the outputs. The final delay is obtained by performing a breadth-first search using the canonical sum and max operations. Defining these canonical operations to be accurate and efficient is the main challenge of SSTA.

It is worth mentioning that all of the above approaches rely on a compact SPICE compatible model. Guard banding and MC simulation can be performed on either the transistor level or gate level. For a transistor-level simulation the model is a must to run SPICE. For gate level simulation, the gates are usually pre-characterized, which is also done in SPICE. SSTA also relies on pre-characterized data for gate delay and power. It is clear that a SPICE compatible model is essential for any type of PV-aware analysis. Such a model is available for CMOS, but if one wishes to explore circuits that use novel devices such as the graphene nano-ribbons (GNR) transistors, the lack of this model becomes the main bottleneck.

This thesis includes two works, which aim to improve the designers' ability to evaluate the impact of PVs on various circuit characteristics. In Chapter 2 a temperature-aware SSTA framework is proposed for silicon transistors. This work aims to capture the effect of PVs on temperature and in turn performance. These effects are interdependent, making the analysis more challenging. In Chapter 3 a compact model for the GNR transistor is developed to enable future PV analysis of this new technology. The next two sections highlight these two works.

1.2 Temperature-Aware Statistical Static Timing Analysis

An HSPICE simulation was performed on a NAND2_X1 gate from the Nangate 45nm Open Cell Library [5] based on the 45 nm predictive technology model [6]. For the high performance process, the delay was 42% larger at 75°C and 92% larger at 125°C. These results are similar to [7], where the 25°C to 125°C delay increase was reported to be 52% for the 65nm

technology node. Even though these results highlight the importance of considering temperature during timing analysis, most of the existing SSTA works do not take this into consideration.

There are a few works such as [8] and [9] that consider temperature and supply variations during timing analysis. The authors of [8] obtain the temperature profile by assuming deterministic power sources and perform deterministic timing analysis. In [9], a similar approach is taken, but the dependence of leakage power on temperature is also considered. Unfortunately, neither of these works considers the effect of PVs on delay and leakage power. In Chapter 2, an algorithm is proposed that evaluates the PDF of a circuit's delay, while accounting for the impact of both PVs and temperature. This is done by first calculating the statistical thermal profile and then performing statistical timing analysis. This algorithm also accounts for the correlation between the gate delay and its temperature, because both the temperature and the delay variations arise from the same source of PVs.

New statistical methodologies are developed to address the lognormal nature of the leakage power and the numerous correlations that exist. The accuracy of this algorithm is verified with MC simulation. In addition, this comprehensive approach is compared to simpler SSTA algorithms to understand the complexity vs. accuracy trade-off for the timing analysis algorithm.

1.3 SPICE Compatible GNRFET Model

Carbon-based nano-materials have emerged as promising successors of CMOS because of their outstanding electrical properties such as high mobility (10X over Si), high current density (10-100X over Cu), low noise and micron-scale mean free path at room temperature. GNRs are particularly appealing due to their planar nature. Since graphene is created in large homogeneous sheets, it can be grown and patterned using standard planar processing techniques [10]. In contrast, nanotubes require a bottom-up method of fabrication [11]-[13].

There have been several publications on device-level modeling and simulation of GNR transistors. The most accurate simulations are obtained with quantum-theory-based NEGF (non-equilibrium Green's function) formalism such as [14], [15], but are very slow. Other works, such

as [16], [17], [18], proposed a semi-classical approach which is faster but produces less accurate results. Both approaches allow detailed device-level simulations but are difficult to scale to circuit-level simulations due to their excessive computation times.

Circuit-level analysis of GNRFETs has been performed by [19] and [20]. Both of these works rely on device-level simulation data, which is either stored in a look-up table or curve-fitted. Unfortunately this approach cannot handle the analysis of PVs, because any change in a process parameter would require one to run the detailed device simulation again. As mentioned earlier, a SPICE compatible model is necessary for this type of circuit exploration. In this work such a model is developed and verified. Then it is used to understand the effect of PVs on gate-level performance of GNR based circuits.

CHAPTER 2

TEMPERATURE-AWARE STATISTICAL STATIC TIMING ANALYSIS

This work was a joint effort with Lu Wan. I developed the temperature aware algorithm and wrote the code that implements the statistical static timing analysis (SSTA) and the MC simulation. Lu characterized the gates and generated the benchmarks for the experiments.

2.1 Introduction

As the process technology continues to scale, the variability of process parameters (PPs) continues to increase. These variations have a significant impact on key performance metrics such as power and propagation delay. According to the International Roadmap for Semiconductors (ITRS) [1], the delay variability was estimated to be 49% in 2010 and will increase to 63% in 2015. In today's technology, the leakage power varies significantly from its nominal value. What is more, this variation is only expected to increase in future technology nodes.

When evaluating the timing of a chip, it is crucial to account for these variations. Early works on timing analysis such as [21] and [22] dealt with these uncertainties by establishing an upper bound. Unfortunately, this approach leads to results that are too pessimistic. Instead, one can obtain the probability density function (PDF) of a circuit's delay by SSTA, which has been a well studied topic [2]-[4],[23],[24], of these, [23] and [24] are path based and enumerate all possible critical paths. Since the maximum number of total paths is exponential to the number of gates, this approach may be computationally expensive. To address this concern, [2]-[4] use block-based SSTA, which consists of traversing the circuit with a breadth first search. It is also believed that the process variations (PVs) of adjacent gates are spatially correlated due to imperfections in chemical-mechanical polishing and lithography. References [2] and [3] handle these correlations by using principal component analysis (PCA). To further improve the accuracy

[25] and [4] develop statistical frameworks that can handle non-Gaussian delay distributions. However, none of these works consider the impact of temperature on delay.

We performed an HSPICE simulation on a NAND2_X1 gate from the Nangate 45nm Open Cell Library [5] based on the 45 nm predictive technology model [6]. For the high performance process, the delay was 42% larger at 75°C and 92% larger at 125°C. These results are similar to [7], where the 25°C to 125°C delay increase was reported to be 52% for the 65nm technology node.

There are a few works such as [8] and [9] that consider temperature and power supply variations during timing analysis. The authors of [8] obtain the temperature profile by assuming deterministic power sources. This profile is later used to adjust the gate delays and improve the accuracy of the timing analysis. In [9], a similar approach is taken, but the dependence of leakage power on temperature is also considered. Unfortunately, neither of these works considers the effect of process variations (PVs) on delay and leakage power. As mentioned earlier, both of these quantities vary significantly, and ignoring this effect can lead to inaccuracies. The leakage power, which has an exponential dependence on process parameters [26], [27], will add significant variation. It is worthy to note, that the total leakage power is expected to triple in magnitude from 2010 to 2015 [1], which will make these statistical considerations even more important in the future.

An algorithm is proposed that evaluates the PDF of a circuit's delay, while accounting for the impact of both PVs and temperature. The statistical thermal profile is calculated and is used to evaluate the statistical timing of the circuit. We also account for the correlation between the gate delay and its temperature, because both the temperature and the delay variations arise from the same source of PVs.

Obtaining an accurate statistical thermal profile in itself is difficult, because the leakage power follows a lognormal distribution and has a strong dependence on temperature. In Ref. [28] both of these issues were addressed to calculate an accurate statistical thermal profile. However, they did not perform timing analysis and did not save the relationship between the temperature distribution and the PVs.

The timing analysis is also a challenge, because the canonical form of the delay is a sum of both normal and lognormal random variables (RV), which are correlated to each other. New statistical methodologies are developed to address this. We also implemented an SSTA engine that only computes the deterministic temperature profile. In this case, the nominal leakage power is used to calculate the temperature distribution and the delay is adjusted accordingly. It was found that this simplified approach can produce reasonable estimates for the 95% and 99% yields. However, the accuracy is reduced for lower target yields such as 85%.

Specifically, the contributions of this work are as follows:

- Accurate statistical thermal profile calculation that considers the thermal-leakage loop and preserves the effect of PVs on the final temperature distribution
- SSTA algorithm that can handle a new canonical form, which is a sum of correlated normal and lognormal variables
- Insight into when it is sufficient to assume deterministic power sources, and when the statistical nature of leakage power must be accounted for

2.2 Preliminaries

2.2.1 Process variations and their impact

Four main sources of variation of a transistor are considered: effective gate length (ΔL), width (ΔW), oxide thickness (ΔT_{ox}), and doping (ΔN_A). The delta corresponds to the deviation from the nominal value. According to [29], these variations can be divided into three components: die to die variation (d2d), spatially correlated variation (cor), and the random component (rand) as shown below:

$$\Delta T_{ox,j} = \Delta T_{ox,d2d,j} + \Delta T_{ox,cor,j} + \Delta T_{ox,rand,j} \quad (2.1)$$

To keep track of the spatial correlations, the die is partitioned into a grid. The gates that are located in the same tile are assumed to be perfectly correlated. For gates in different tiles this

correlation decreases with distance. To keep track of these correlations, principal component analysis (PCA) is used ([3], [25], [30]) to express a set of correlated variables as a new set of independent random variables. Equation 2.2 shows an example for $\Delta T_{ox,j}$. Here, $\Delta T_{ox,j}$ corresponds to the variation in oxide thickness at tile j, R stands for the purely random component, $T_{ox,ji}$'s are the constant coefficients, and n is the number PCs. Finally, $PC_{Tox,i}$'s are the standard normal (SN) distributions that represent the principal components (PCs) of ΔT_{ox}; note that this PC vector is the same for all of the tiles.

$$\Delta T_{ox,j} = \sum_{i=1}^{n} (T_{ox,j,i} \times PC_{Tox,i}) + T_{ox,j,R} \times R \tag{2.2}$$

Figure 2.1 qualitatively shows how PVs affect power, temperature, and delay. First, PVs affect leakage power, which in turn changes the temperature. The rise in temperature increases the leakage power, which further heats up the die. The gate delay is affected by both temperature and PPs such as ΔL_{eff}, ΔW, ΔT_{ox}, and ΔN_A. Note that the temperature profile is also a function of these same PPs. Thus the gate's delay variation caused by temperature and the variation caused directly by the PPs are correlated. In order to account for this effect during the timing analysis, it is important to express the statistical temperature profile in terms of the original PVs.

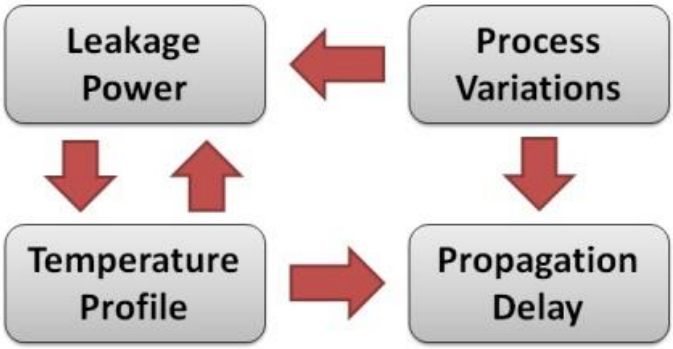

Figure 2.1: Impact of Process Variations

2.2.2 Gate leakage power modeling

In this work, the leakage power of a gate is assumed to be an exponential function of ΔL and ΔT_{ox} and a quadratic function of temperature according to [31]. The exponential dependence on process parameters is modeled as follows:

$$P'_{leak,g} = P_{leak,nom} \times \exp(b \times \Delta T_{ox} + c \times \Delta L) \quad (2.3)$$

where $P_{leak,nom}$ is the leakage power without PVs at 0°C, while $P'_{leak,g}$ is the leakage power at 0°C with PVs. Constants b and c are the sensitivity coefficients to the respective PPs. Equation 2.4 models the complete leakage power of a gate ($P_{leak,g}$) by multiplying $P'_{leak,g}$ by the temperature dependent term. Here, T_g is the temperature of the gate and a_1 and a_2 are the fitting parameters.

$$P_{leak,g} = P'_{leak,g} \times \left(1 + a_1 \times T_g + a_2 \times T_g^2\right) \quad (2.4)$$

2.2.3 Gate delay modeling

The gate delay (D_g) is assumed to be a linear function of the PPs similarly to [3] and [30] as shown in Equation 2.5. ΔL, ΔW, ΔT_{ox}, ΔN_A are PVs discussed in section 2.2.1 and α, β, γ, and ε are the delay sensitivities to these respective PPs. $D_{nom,g}$ is the gate delay at nominal temperature and no PVs, T_g is the gate temperature, and t_1 is the sensitivity of the gate delay on T_g. Essentially, an extra component is added to account for the temperature effect.

$$D_g = D_{nom,g} + \alpha \times \Delta L + \beta \times \Delta T_{ox} + \gamma \times \Delta W + \epsilon \times \Delta N_A + t_1 \times Tg \quad (2.5)$$

2.2.4 Thermal modeling

To convert the power profile into a temperature profile, the concept of a thermal admission matrix is used similarly to [28]. Every entry of A, A_{ij}, corresponds to the temperature increase at tile i caused by a power source at tile j as shown in Equation 2.6. Thus the temperature at every tile is a weighted sum of tile powers and can be computed with Equation

2.7. In this equation T_{mx1} is the vector of all the tile temperatures, P_{mx1} is the vector of all the tile powers, A is the admission matrix, and m is the number of tiles in the grid.

$$A_{i,j} = \frac{\Delta T_i}{\Delta P_j} \tag{2.6}$$

$$T_{m\times 1} = A_{m\times m} \times P_{m\times 1} \tag{2.7}$$

2.2.5 SSTA review

A common way to find the PDF of a circuit's delay is to perform a PERT-like traversal of a timing graph, while propagating the statistical delay distribution through the circuit [3], [30]. To aid this traversal, two statistical canonical operations are defined: "max" and "sum". Figure 2.2 illustrates the traversal of a NAND gate using these operations. D_{GA} is defined as the time it takes for output to change after input A has changed and D_{GB} is defined analogously. The first step is to "sum" D_A and D_{GA} as well as D_B and D_{GB}. Both of these represent the statistical delay through the potential timing path. Then results are "maxed" to obtain D_C, which is the delay distribution at the output of the NAND gate. If canonical operations "max" and "sum" are defined, the delay of any digital circuit can be computed with the above procedure.

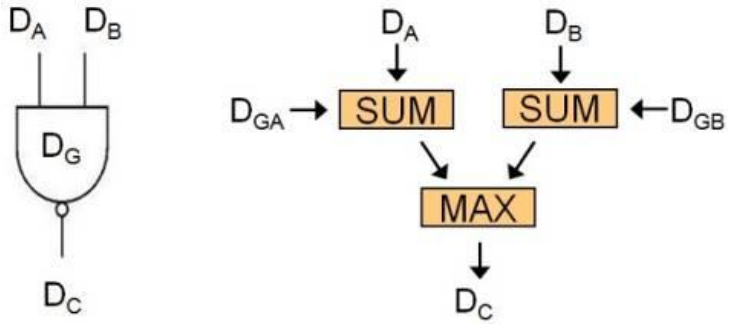

Figure 2.2: Operations on a NAND gate

For this approach to work, it is critical that all of the delays are expressed in the same canonical form on which "max" and "sum" can operate. To convert the gate delays to this form,

we first express all of the process parameters in terms of the PCs as shown previously in (2). Then these are plugged into (5) to obtain:

$$D_g = D_{nom,g} + \alpha\left(\sum_{i=1}^{n} L_{j,i} \times PC_{L,i} + L_{j,R} \times R\right) + \\ +\beta\left(\sum_{i=1}^{n} T_{ox,j,i} \times PC_{Tox,i} + T_{ox,j,R} \times R\right) \ldots \quad (2.8)$$

Only ΔL and ΔT_{ox} are listed in the equation above, but this principle is extended to also account for ΔW and ΔN_A. Notice that ΔL and ΔT_{ox} are both expressed in terms of different PC vectors and different coefficients. To simplify the notation, PC_{Tox}, PC_L, PC_W, and PC_{Na} are lumped into one vector denoted PC_i's. Their corresponding constant coefficients are multiplied by the sensitivity coefficients α, β, γ, and ε to form the a_i's i:

$$D_A = D_{nom,A} + \sum a_i \times PC_i + a_R \times R \quad (2.9)$$

Here, a_i's are the coefficients and PC_i's are the principal components. This formulation allows the delay of any node to be explicitly expressed in terms of the constant coefficients a_i's.

The "max" and "sum" operations are defined to take in two delays D_A and D_B in the canonical form as shown in Equation 2.9 and output a delay D_C in the same form by computing D_C's constant coefficients c_i's. For the sum operation c_i's can be found with equations from [3]:

$$D_{c,nom} = D_{a,nom} + D_{b,nom} \quad (2.10)$$

$$c_i = a_i + b_i \quad (2.11)$$

$$c_R = \sqrt{a_R^2 + b_R^2} \quad (2.12)$$

The max operation utilizes Clark's method [32] to calculate the mean, variance, and P_A which is defined as the probability that D_A is larger than D_B. Once these values are known the D_c's coefficients are set using the following expressions [25]:

$$D_{C,nom} = mean(D_C) \tag{2.13}$$

$$c_i = P_A \times a_i + (1 - P_A) \times b_i; \tag{2.14}$$

$$c_R = \sqrt{var(D_C) - \sum_1^n c_i^2} \tag{2.15}$$

2.3 Algorithm Description

2.3.1 Overview

In order to account for the impact of the statistical thermal profile on delay, a new flow was flows as shown in Figure 2.3. First, all of the process parameters are expressed in terms of PCs using PCA. The temperature profile is a lognormal because it is a weighted sum of correlated gate leakage powers, which are also lognormal. Note that the correlated component dominates the random component, because it adds up, while the random mismatch tends to cancel out. As shown in section 2.3.2, it is critical to express the temperature profile in terms of PCs to enable the timing engine in the next steps. This has not been done before, so a new frame work was developed, which will be presented in section 2.3.2. Once the temperature is calculated, the gate delays are set to canonical form and SSTA is performed to obtain the PDF of the final delay. Since T_g is a lognormal, the temperature dependent term of the D_g is also a lognormal. Thus the propagated PDF is a sum of a normal RV and a lognormal RV. To address this challenge, a new canonical form is defined (discussed in section 2.3.4) for the delay, and new canonical operations "max" and "sum" are developed to handle it. In the following subsections, each of these steps will be discussed in further detail.

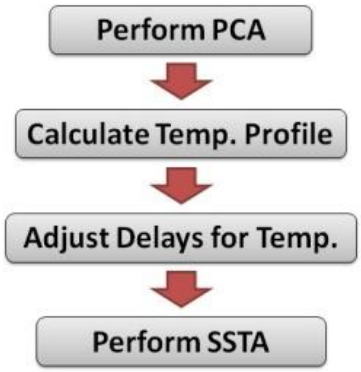

Figure 2.3: Algorithm Overview

2.3.2 Statistical thermal profile in terms of process parameters

Figure 2.4 shows the high level algorithm for calculating the temperature profile. First, the temperature due to deterministic dynamic power (T_{det}) is calculated. This information is used to set the initial leakage power. Then, the tile power is calculated by adding up the power of all the gates located in the given tile. Once the tile powers are known, the statistical thermal profile is obtained, which is then used to update the leakage powers. We iterate through the loop until convergence is reached. Note that both gate leakage power ($P_{leak,g}$) and the temperature increase caused by leakage power T_{leak} are lognormal RVs. To perform the calculations shown in Figure 2.4, one must be able to add and multiply the lognormal variables, while preserving correlations. To accomplish this, "sum" and "multiply" operations are defined whose inputs and outputs are in lognormal canonical form as shown below:

$$L_A = \exp(X_A)\,;\; X_A = \; p_{con} + \sum_{i=0}^{n*m} p_{a,i} \times PC_i + p_R \times R \qquad (2.16)$$

Any lognormal in canonical form (L_A) is expressed in terms of X_A, which is a normal distribution. p_{con} is the mean of X_A, while $p_{a,i}$'s and p_R are constant multipliers.

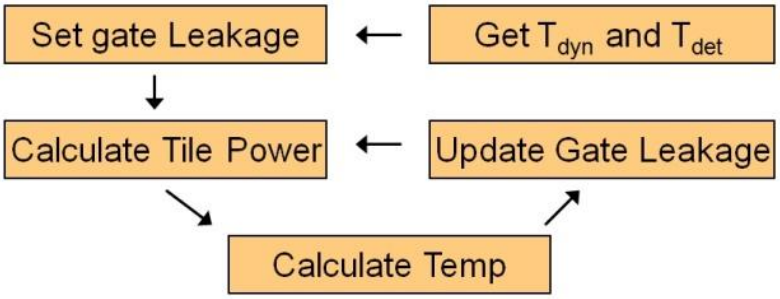

Figure 2.4: Computing the statistical thermal profile

In order to preserve the relationship between a given lognormal and the underlying PVs, X_A is expressed as a weighted sum of PCs and the random component R. With this definition, knowing p_{con}, $p_{a,i}$'s, and p_R is sufficient to describe any lognormal. As long as $P_{leak,g}$, T_{leak}, and all of the intermediate values are expressed in lognormal canonical form shown in Equation 2.10, the statistical thermal profile can be computed using the flow shown in Figure 2.4.

In the following sections each step will be explained in more detail and the procedure for performing "lognormal sum", "lognormal multiply", and "multiply constant" will be discussed in section 2.3.2.5. Also, note that designs with high power density can experience thermal runaway. In this scenario, temperature and leakage enter a positive feedback loop where both temperature and leakage continue to rise, which leads to thermal chip failure. When such a condition is simulated in the proposed algorithm, the loop in Figure 2.4 will diverge and the algorithm would return error.

2.3.2.1 Temperature profile from dynamic power

Since the temperature is a weighted sum of powers, we can separate it into three components:

$$T_i = T_{amb,i} + T_{dyn,i} + T_{leak,i} = T_{det,i} + T_{leak,i} \quad (2.17)$$

where T_i represents the temperature at tile i, T_{amb} is the ambient temperature, T_{dyn} is the temperature increase due to dynamic power, and T_{leak} is the temperature increase from the leakage power. According to [33], the dynamic power is not very sensitive to PVs, so its

nominal value is used for computations. Also note that the time constant for temperature change is much larger than the input vector toggle rate (ms vs. ns). Thus it is sufficient to use the average dynamic power without worrying about temporary changes in temperature [28]. With this assumption, T_{dyn} can be computed using Equation 2.7. Then, it is lumped together with T_{amb} to form T_{det}. In previous work [28], the T_{det} and T_{leak} are combined together into one lognormal. This adds error to the calculation, since a lognormal (T_{leak}) added to a constant (T_{det}) is no longer a lognormal. By keeping these separate the accuracy is preserved. This operation separates the statistical component of temperature from the deterministic component of temperature to improve the accuracy.

2.3.2.2 Setting leakage power in canonical form

To avoid adding T_{det} and T_{leak} (lognormal + constant) the expression for leakage power is manipulated to only be in terms of T_{leak}. After the temperature is separated into two components, the leakage power expression is expanded to:

$$\begin{aligned} P_{leak,g} &= P'_{leak,g} \times (1 + a_1(T_{det} + T_{leak}) + a_2(T_{det} + T_{leak})^2 \\ &= P_{leak,g,adj} \times (1 + a'_1 \times T_{leak} + a'_2 \times T_{leak}{}^2) \end{aligned} \tag{2.18}$$

At this step, T_{det} is already known and is thus a constant. Using algebraic manipulations the second half of Equation 2.18 is obtained. This removes the T_{det} term and simplifies the expression. Essentially the leakage power is re-characterizing around T_{det} instead of 0°C. $P_{leak,g,adj}$, a_1' and a_2' can be found using the expressions below:

$$P_{leak,g,adj} = P_{leak,g} \times (1 + a_1 \times T_{det} + a_2 \times T_{det}{}^2) \tag{2.19}$$

$$a'_1 = \frac{a_1 + 2a_2 \times T_{det}}{1 + a_1 \times T_{det} + a_2 \times T_{det}^2} \tag{2.20}$$

$$a_2' = \frac{a_2 \times T_{det}}{1 + a_1 \times T_{det} + a_2 \times T_{det}^2} \tag{2.21}$$

Now we are ready to set the initial leakage power. Initially, T_{leak} is 0°C so the leakage power simplifies to $P_{leak,g,adj}$. Writing this in terms of the PPs, the expressions below are obtained:

$$\begin{aligned} P_{leak,g,adj} &= P_{leak,g,nom} \times \exp(b \times \Delta L + c \times \Delta T_{ox}) \times \\ &\times (1 + a_1 \times T_{det} + a_2 \times T_{det}^2) \end{aligned} \tag{2.22}$$

Next, both ΔL and ΔT_{ox} have to be expressed in terms of their PCs. To simplify the notation one can perform a step similar to the one presented in section 2.2.5 for gate delay. Once again the goal is to express the leakage power in lognormal canonical form show in Equation 2.16. All of the PCs and their coefficients are lumped into one vector of PC_i's and $p_{a,i}$ respectively. The constant terms in Equation 2.22 are combined together and brought to the exponent to form p_{con}.

2.3.2.3 Computing the temperature

Once the P_{leak}'s are known at every gate, the tile powers are computed with the "lognormal sum" operation, which will be explained in section 2.3.2.5. To get the temperature, a weighted sum of the tile powers must be performed. This is accomplished by scaling the tile power by their respective weights using the "multiply constant" operation and then adding up the results with "lognormal sum".

2.3.2.4 Updating the leakage power

Once the statistical thermal profile is calculated, the leakage powers must be updated according to Equation 2.18. This expression can be quickly computed after the sum and multiply operations are available. After the leakage power is updated, the temperature profile can also be recalculated. This process is repeated until convergence is reached.

2.3.2.5 Statistical operations for lognormals

To perform the steps discussed in section 2.3.2.2 and section 2.3.2.4, one must define an effective "lognormal sum", "lognormal multiply", and "multiply constant". When explaining these operations, it is assumed that there are two input lognormals L_A and L_B, and an output lognormal L_C. X_A, X_B, and X_C refer to the normal distributions that are in the exponent of L_A, L_B, and L_C respectively. $p_{a,i}$'s, $p_{b,i}$'s, and $p_{c,i}$'s refer to the constant coefficients that define X_A, X_B, and X_C. First, consider the sum operation. The mean and variance of X_A and X_B as well as the covariance between X_A and X_B are computed first. This is simple, since they are just a sum of normal RVs. Using the properties of lognormals, the means of L_A (μ_A) and L_B (μ_B), variances of L_A (σ_A^2) and L_B (σ_B^2), and the covariance of L_A and L_B (σ_{AB}) can be found. This allows one to find the mean (μ_C) and variance (σ_C^2) of L_C. Finally, p_{con}, $p_{a,i}$'s and p_R can be set using Equations: 2.23, 2.24, and 2.25. Multiplying two exponential functions is equivalent to adding their exponents. Thus performing the "multiply" operation boils down to adding two normal distributions X_A and X_B to form X_C. This can be accomplished with the normal sum operation discussed in section 2.2.5. For the multiply constant operation, we multiply L_A by a constant z to form L_B. This is done by bringing the exponent z to the exponential. Essentially the constant term in the exponent $p_{b,con}$ is modified by ln(z) as shown in Equation 2.26. The rest of the coefficients in L_B remain the same as L_A.

$$p_{c,i} = ln\left(\frac{u_A \times \exp(p_{a,i}) + u_B \times \exp(p_{b,i})}{u_A + u_B}\right) \tag{2.23}$$

$$p_{c,const} = \ln(u_C) - \frac{1}{2} \times \ln\left(1 + \frac{\sigma_{Lc}^2}{u_{Lc}^2}\right) \tag{2.24}$$

$$p_{c,n+1} = \sqrt{\ln\left(1 + \frac{\sigma_{Lc}^2}{u_{Lc}^2}\right) - \sum_1^n p_{c,i}^2} \tag{2.25}$$

$$L_B = z \times L_A \;\leftrightarrow\; p_{b,con} = p_{a,con} + \ln(z) \tag{2.26}$$

2.3.3 Adjusting gate delay based on temperature

Once the temperature profile is computed, the delay of each gate is set using Equation 2.27. This is the same expression as Equation 2.5, but the temperature term is expanded into T_{leak} and T_{det} terms. The deterministic temperature term can be multiplied by t_1 and lumped together with $D_{nom,g}$ to form $D_{nom,g}^{'}$. Thus one is left with the normal term N_A, coming from the PVs, and a lognormal term L_A caused by T_{leak}. By writing these in terms of the PC vectors, D_g can be expressed in canonical form shown in Equation 2.28. This is the final canonical form that will be used for timing analysis. Note that the N_A and X_A are correlated, since they are expressed in terms of the same PC vector.

$$D_g = D_{nom,g} + \alpha \times \Delta L_{eff} + \beta \times \Delta T_{ox} + \\ + \gamma \times \Delta W + \epsilon \times \Delta N_A + t_1 \times (T_{det} + T_{leak}) \tag{2.27}$$

$$D_A = N_A + L_A = N_A + \exp(X_A) = D_{nom,a}^{'} + \sum a_i \times PC_i + a_R \times R + \\ + \exp\left(p_0 + \sum_{i=0}^{n*m} p_{a,i} \times PC_i + p_R \times R\right) \tag{2.28}$$

2.3.4 Timing graph traversal

To obtain the final delay distribution, the timing graph is traversed and the canonical form is propagated similarly to section 2.2.5. This time the canonical form shown in Equation 2.28 includes a correlated normal and lognormal term, which makes the task more challenging. To enable the SSTA engine, new statistical operations are developed for "Sum" and "Max", which are described in the following sections.

2.3.4.1 Delay sum operation

The sum operation is relatively straight forward. The normal portion of the distribution can be added in the way described in section 2.2.5. The lognormal portion can be added using the lognormal sum operation described in section 2.3.2.5.

2.3.4.2 Delay max operation

The N's and X's are expressed in terms of the same PCs as shown in Equation 2.28. The coefficients of N_A, N_B, and N_C are a_i's, b_i's and c_i's respectively. The coefficients of X_A, X_B, and X_C are $p_{a,i}$'s, $p_{b,i}$'s and $p_{c,i}$'s respectively. Note that all four of these RVs are correlated as shown in the left picture in Figure 2.5. The arrows indicate that a correlation exists. Solving this case directly with discretization will require discretization in 4 dimensions and will be computationally expensive. To keep computation time under control an approximation is developed that only requires discretization in 2 dimensions. In short, the input distributions are redefined so that the lognormal and normal components are independent. Then the lognormal RV is discretized and Clark's algorithm [32] is used.

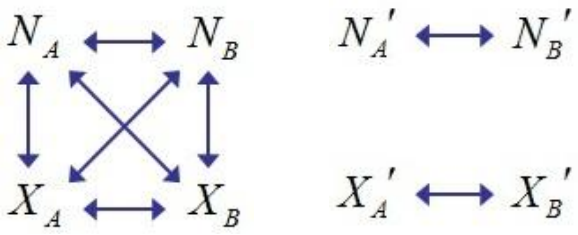

Figure 2.5: (left) All correlations present, (right) reduced correlations

First, two new delays, D_A' and D_B', are defined as shown in Equation 2.29. These are defined such that the max of D_A and D_B is closely approximated by the max of D_A' and D_B'. N_A', N_B', X_A', and X_B' are chosen such that there are no correlations between the normal and the lognormal terms as illustrated in Figure 2.5.

$$D_A' = N_A' + \exp(X_A'); \;\; D_B' = N_B' + \exp(X_B') \tag{2.29}$$

To ensure that the max (D_A', D_B') matches the max (D_A, D_B) closely, we impose the following conditions:

- mean(D_A') = mean(D_A); mean(D_B') = mean(D_B)
- var (D_A') = var(D_A); var(D_B') = var(D_B)
- covariance (D_A', D_B') = covariance (D_A, D_B)

It is also important to keep the ratio of the normal component to the lognormal component the same. To meet these requirements, 3 adjustment coefficient c_1, c_2, and c_3 are defined as shown below:

$$c_1 = \frac{var(N_A) + var(L_A) + 2cov(N_A, L_A)}{var(N_A) + var(L_A)} \tag{2.30}$$

$$c_2 = \frac{var(N_B) + var(L_B) + 2cov(N_B, L_B)}{var(N_B) + var(L_B)} \tag{2.31}$$

$$c_3 = \frac{cov(N_A, N_B) + cov(L_A, L_B) + cov(N_A, L_B) + cov(N_B, L_A)}{cov(N_A, N_B) + cov(L_A, L_B)} \tag{2.32}$$

With these correction factors in place the different variances and covariances are set according to equations below:

$$var(N_A') = c_1 var(N_A);\ \ var(L_A') = c_1 var(L_A) \tag{2.33}$$

$$var(N_B') = c_1 var(N_B);\ \ var(N_B') = c_1 var(N_B) \tag{2.34}$$

$$cov(N_A', N_B') = c_3 \times cov(N_A, N_B) \tag{2.35}$$

$$cov(L_A', L_B') = c_3 \times cov(L_A, L_B) \tag{2.36}$$

In the expressions above the L's represent the lognormal components of the delay. For example L_A, is exp (X_A). Once the means and variances of L_A' and L_B' are known the mean and SD of X_A' and X_B' can be found using the properties of lognormals. The correlation of X_A' and X_B' can also be computed from the covariance of L_A' and L_B'.

Once D_A' and D_B' are defined, the mean, SD, and P_A of D_C' can be found. Note that the mean, SD, and P_A of D_C will be approximated by the values obtained for D_C'. The procedure for finding the mean of D_C', denoted with μ_C is demonstrated. Note that the second moment (m2) and P_A can be obtained analogously. First, the conditional mean $\mu_{C,i,j}$ is defined with Equation

2.37. Essentially it is the value of μ_C if X_A' equaled c_i and X_B' equaled c_j, where c_i and c_j are constants. In that case, the max can be expressed with Equation 2.38. Notice that the exp (c_i) and exp (c_j) are just constants and can be lumped together with the mean of N_A', and N_B' respectively. Now Clark's algorithm can be applied to find $\mu_{C,i,j}$. The actual mean of D_C' can be obtained by multiplying the conditional means by their respective probabilities and adding up the results as shown in Equation 2.39.

$$u_{c,i,j} = u_c|(X_A^{'} = c_i \ \& \ X_B^{'} = c_j) \tag{2.37}$$

$$u_{c,i,j} = mean(\max(N_A^{'} + \mathrm{e}^{c_i}, N_B^{'} + \mathrm{e}^{c_j})) \tag{2.38}$$

$$u_c = \sum u_{c,i,j} \times P(X_A^{'} = c_i \ \& X_B^{'} = c_j) \tag{2.39}$$

Note that the mean and SD of X_A and X_B as well as their correlation are known. Thus the expression for the bivariate normal distribution can be used to calculate the probability that X_A' equals c_i and X_B' equals c_j. From the experiments it was found that discretizing 7 points in each of the two dimensions provides a good balance between run-time and accuracy. This procedure is repeated for P_A and the second moment, which is used to find the variances.

Once the mean, variance, and P_A are obtained, D_C must be expressed back into canonical form. First, the correlated component of N_C is computed by setting the c_i's with Equation 2.14. Then the lognormal component L_C is computed as a weighted sum of L_A and L_B using Equation 2.40. This can be performed with the lognormal sum operation described in 2.3.2.5. Once L_C and the c_i's are set, the $D_{nom,C}$ and the c_R are chosen such that the total mean and variance of the D_C matches the previously calculated values.

$$L_c = P_A \times L_A + (1 - P_B) \times L_B \tag{2.40}$$

2.4 Experimental Results

To evaluate the described framework, the proposed flow was implemented in C++ and compared to Monte Carlo (MC) simulation to evaluate the accuracy. VPR [34] benchmarks were synthesized using the Nangate 45nm library [6] and the cells were placed using Cadence Encounter. To ensure that enough power is dissipated and the thermal effects are observed, we targeted die that were 1mm by 1mm. Small benchmarks were multiplied to achieve sufficient total power dissipation. The publicly available tool ISAC2 [35] was used to obtain the thermal admission matrix. The lateral dimensions of the default chip configuration were scaled down to obtain the 1mm x 1mm die size. The nominal power was calculated using the Synopsis Design Compiler by running different input vectors at 1 GHz frequency and a switching activity factor of 0.15. We set the dynamic power to the average that was obtained from the random input vectors. It was assumed that the nominal leakage power makes up 33% of the total power similarly to [28].

To model spatial correlations the die was portioned into a 20 by 20 grid. The total amount of variation in the process parameters was set such that $3\sigma/\mu = 20\%$. This was divided into 25% d2d, 55% correlated, and 20% random based on [29]. The correlations between 2 tiles follows a diminishing function of exp(-b*d), where d is the distance between the tiles and b is a constant. The same grid was used for PCA and thermal analysis. To obtain the sensitivity of the leakage power and delay to different PPs, HSPICE simulations were ran and the results were curve fitted with MatLab. The MC simulations were performed by iteratively generating the process parameters and performing thermal and timing analysis. Under certain combinations of process parameters, thermal run-way was observed and the numerical calculations resulted in very high temperatures. These were treated as failed chips and were subtracted from the total yield.

Table 2.1 shows the benchmark circuits that were used along with their number of gates, average temperatures, and the run-time of the tool.

Table 2.1: Benchmark Circuit Description

#	Benchmark Circuit	# of gates	Mean Temp	Run - time(s)
1	sv_chip0_hierarchy_no_mem	42478	36	510
2	cf_cordic_v_18_18_18_X4	67677	43	499
3	cf_fir_24_16_16_16	48791	42	86
4	sv_chip1_hierarchy_no_mem	78554	40	1941
5	max1_X9	48655	32	84
6	paj_boundtop_hierarchy_ no_mem_X6	20092	33	257
7	paj_raygentop_hierarchy_ no_mem_X4	91360	41	7627
8	cf_cordic_v_8_8_8_X16	67753	40	1561
9	des_perf_X2	49055	42	131
10	oc54_cpu_X16	148412	40	636

The X in the benchmark name implies that the final circuit was obtained by duplicating the original VPR benchmark X times. One may notice that the run-time is quiet long. However, these benchmark circuits are much bigger than the ones used in previous works such as [3] and [25]. The run-time was compared to traditional SSTA and was found to be 2-3 times longer. This is reasonable, since an extra step was added for temperature computation and the canonical form of the delay contains a normal and a lognormal term. In Table 2.2, the accuracy of this approach is verified against MC simulation. The mean, SD, and the performance yield at 85% and 95% were considered. A 95% performance yield implies that 95% of the fabricated units will be faster than the given delay target. The % error reported in each category is calculated using Equation 2.41.

$$\%error = \frac{Analy - MC}{MC} \times 100\% \tag{2.41}$$

The average error reported is obtained by taking the average of the absolute errors. This ensures that the positive and negative errors will not cancel each other out and lead to a false sense of high accuracy. One can see that the proposed approach is quite accurate with a mean

and SD error of 0.95% and 3.45% respectively. The yield error is also low for both the 85% and 95% case: 0.79% and 0.91% respectively.

Table 2.2: Error of Temperature-Aware SSTA Compared to MC-Simulation

circuit #	Monte-Carlo Simulation				SSTA with Stat. Temp (our)				% Error			
	mean (ps)	SD (ps)	yield (ps)		mean (ps)	SD (ps)	yield (ps)		mean	SD	yield	
			85%	95%			85%	95%			85%	95%
1	1798	147.1	1955	2051	1813	140.9	1958	2047	0.84	-4.20	0.16	-0.16
2	2878	232.3	3133	3307	2887	216.5	3111	3260	0.32	-6.79	-0.70	-1.43
3	2426	181.7	2607	2736	2442	180.1	2629	2750	0.68	-0.86	0.86	0.52
4	1812	147.2	1968	2068	1810	147.3	1970	2067	-0.11	0.07	0.09	-0.05
5	3244	316.8	3592	3800	3279	305.1	3596	3787	1.09	-3.67	0.12	-0.32
6	2549	242.4	2793	2945	2618	238.6	2866	3015	2.74	-1.57	2.62	2.37
7	3142	293.2	3436	3638	3163	274.5	3449	3631	0.67	-6.38	0.37	-0.19
8	2997	254.7	3260	3446	2977	242.6	3230	3392	-0.64	-4.75	-0.93	-1.57
9	2866	222.9	3115	3286	2879	216.4	3104	3250	0.47	-2.92	-0.35	-1.09
10	3659	288.9	3957	4154	3729	279.4	4025	4210	1.94	-3.29	1.70	1.36
Average abs. error:									**0.95%**	**3.45%**	**0.79%**	**0.91%**

Two simpler versions of SSTA were also evaluated. Table 2.3 shows the accuracy of these approaches. Once again, the error is calculated using Equation 2.41 for the mean, SD, and the target yields. The first approach, labeled as "SSTA at nominal temperature", corresponds to the traditional SSTA [3], which assumes room temperature of 25°C for delay computations. For the "SSTA with deterministic temperature", the nominal leakage power was used to compute a deterministic temperature profile. Then, the gate delays were adjusted according to the deterministic temperature of the tiles. One can observe that performing SSTA at room temperature results in a large error in both mean and SD (7% and 20.9% respectively). This is expected and shows the impact of temperature on the statistical delay of a circuit.

Table 2.3: Accuracy of Simplified SSTA Algorithms

circuit #	% Error against Monte-Carlo Simulation							
	SSTA at nominal temperature				SSTA with deterministic temperature			
	mean	SD	85% yield	95% yield	mean	SD	85% yield	95% yield
1	-4.39	14.34	-3.15	-2.70	-1.78	14.34	-0.75	-0.41
2	-8.44	28.45	-6.00	-5.50	-4.07	28.45	-1.98	-1.69
3	-8.16	26.66	-5.36	-4.75	-3.40	26.66	-0.93	-0.53
4	-7.45	17.86	-5.64	-5.12	-5.46	17.86	-3.81	-3.38
5	-3.39	15.40	-2.18	-1.70	-1.50	15.40	-0.47	-0.09
6	-5.34	12.87	-3.43	-2.80	-3.22	12.87	-1.50	-0.97
7	-7.86	19.00	-5.20	-4.67	-4.34	19.00	-1.98	-1.63
8	-8.81	27.26	-5.84	-5.24	-5.02	27.26	-2.36	-1.95
9	-8.52	27.73	-6.35	-5.98	-4.44	27.73	-2.59	-2.42
10	-7.47	19.21	-5.40	-4.86	-3.77	19.21	-1.98	-1.60
average:	**6.98%**	**20.88%**	**4.86%**	**4.33%**	**3.70%**	**20.88%**	**1.84%**	**1.47%**

When deterministic temperature is considered, the mean accuracy improves, but the SD is still inaccurate. This is because accounting for the deterministic temperature profile simply shifts (increases the mean) the PDF of the final delay. Also note that the standard deviation of both of these approaches overestimates the SD. This suggests that accounting for the statistical temperature profile reduces the SD of the final delay distribution. This can be explained by the negative correlation between the temperature dependent delay term and the rest of the delay. Essentially, PPs such as ΔL and ΔT_{ox} affect delay and leakage in opposite directions. For example as the channel length is reduced, the delay becomes faster, but the leakage power increases. The same is true with the oxide thickness. One can observe that the SSTA with deterministic temperature profile can give a reasonable yield estimate: 1.84% and 1.47% error for the 85% and 95% yield respectively. This may be surprising, but note that the SSTA with deterministic temperature underestimates the mean, but overestimates the SD. For a Gaussian distribution the performance yield can be calculated with Equation 2.42, where u and σ are the

mean and SD of the final timing curve and γ is a multiplier which is dependent on the yield target. For example, γ equals 1.645 for the 95% yield and equals 1.04 for the 85% yield.

$$yield = u + \gamma \times \sigma \tag{2.42}$$

Since the mean is underestimated and the SD is overestimated, the two errors tend to cancel out and improve the accuracy of the yield calculation. Table 2.4 compares the error from an SSTA algorithm that considers temperature statistically and an SSTA algorithm that only computes the deterministic temperature profile. We report the accuracy improvement that results from accounting for statistical power sources.

Table 2.4: Comparing Error of SSTA with Statistical vs. Deterministic Temperature for Different Yields

Yield target	Average % error of SSTA against MC Simulation		Accuracy improvement with stat. temp
	det. Temp	stat. temp	
99%	1.38%	1.36%	1.01x
95%	1.47%	0.91%	1.62x
90%	1.59%	0.79%	2.01x
85%	1.84%	0.79%	2.33x

One can see that the improvement is higher for lower yields and lower for higher yields. This can be explained by the fact that the γ in Equation 2.42 is higher for higher yield targets. Thus the mean and SD errors cancel out better.

To obtain a better intuition, one can also think of the circuit directly. In general, if a circuit has very poor performance it is not likely to have very high leakage power, because the process parameters will affect delay and leakage differently. For example if channel length is increased, the delay will go up, but the leakage will be reduced. Thus when one considers the 99% yield target, the slowest 1 % of the dies will have low leakage power. However, if one looks at the slowest 15 % (targeting 85% yield), one is more likely to find some die that have high leakage power that resulted from random variations. For these cases it is important to

account for the statistical temperature profile. One may think that considering the temperature as an independent variable may give good results and have a reasonable run-time. Unfortunately, the temperature and delay have a strong negative correlation, which significantly reduces the final SD. Thus this approach would overestimate the SD more than the case with the deterministic temperature profile, leading to larger inaccuracies.

CHAPTER 3

GNRFET COMPACT MODEL FOR TECHNOLOGY EXPLORATION

3.1 Introduction

Although conventional CMOS devices have prevailed in the semiconductor industry for decades, it has been increasingly difficult to keep up with Moore's law due to the various challenges imposed by the extremely small feature sizes, including increased wire resistivity, significant mobility degradation, and large dopant fluctuations. Meanwhile, carbon-based nano-materials have emerged as promising successors of CMOS because of their outstanding electrical properties and potentially large integration capabilities through new fabrication techniques [10]-[13]. The most studied carbon-based nano-materials today are carbon nanotubes (CNTs) and graphene nano-ribbons (GNRs). They have demonstrated high mobility (10X over Si), high current density (10-100X over Cu), low noise and micron-scale mean free path at room temperature [37]-[40]. They are also exceptionally robust structures and have high thermal conductivity (10X over Cu) [12]. Although CNTs have slightly better electrical properties than GNRs, GNRs are considered more controllable and scalable in terms of fabrication due to the planar nature of graphene. Since graphene is created in large homogeneous sheets, it can be grown and patterned using standard planar processing techniques [11].

To date, there have been several publications on device-level modeling and simulation of GNR transistors. The most accurate simulations are obtained with quantum-theory-based NEGF (non-equilibrium Green's function) formalism such as [14], [15], but are very slow. The effect of edge roughness is also evaluated in [14]. It is predicted that perfectly smooth GNRs will not be fabricated and instead a given GNR will vary in width. Other works, such as [16], [17], [18], proposed a semi-classical approach which is faster, but produces less accurate results. Both approaches allow detailed device-level simulations but are difficult to scale to circuit-level simulations due to their excessive computation times. In order to enable true exploration of graphene based technology, a SPICE compatible model is required. This would allow designers

to input various GNRFET settings (length, width, oxide thickness) and quickly evaluate the performance of a desired circuit.

In [19], a circuit-level simulation framework is employed for technology exploration of Schottky barrier (SB) GNRFET circuits. First, GNRFETs are simulated at the device level. The dc I-V (current vs. gate and drain voltage) and Q-V (channel charge vs. gate and drain voltage) characteristics for GNRFETs are obtained from solving NEGF with 3-D Poisson's equation. Then, the I-V and Q-V data are used to build a look-up table to support the circuit-level simulation engine. Basic digital circuits such as inverters, ring oscillators, and latches are analyzed in this framework. Unfortunately this model is difficult to scale or extend, because a detailed device simulation is necessary for every set of device settings. A similar framework was used to evaluate the graphene tunneling FET in [36], where the variation in width and doping level was considered. Thus this approach is not as effective as a SPICE-compatible model. Recently, a compact physics-based simulation framework has been proposed in [20] and the accuracy was validated with NEGF simulations. However, multiple parameters in this model were obtained by curve-fitting. Again this model is not a true compact model. A different semi-analytical model was developed for SB GNRFETs in [18], which would allow the user to obtain the IV data for any given device settings. Unfortunately this model relies on the computation of numerical integrals and cannot be directly used for SPICE-level circuit simulations.

Essentially, one set of works can support any device setting, but is computationally expensive and cannot be used for SPICE simulation [16], [17], [18]. Another set of works characterize a single device by curve-fitting or a look up table and use it for circuit-level simulation [19], [20]. The author bridges the gap between these sets of works to provide a parameterized SPICE compatible circuit model, which allows a designer to quickly simulate a circuit with any GNRFET settings. We plan to release this model to aid designers in exploring graphene based circuits.

In addition most of the existing works focus on one aspect of graphene circuit design, whether it is the device [15], [18] or the graphene interconnect [41], but do not discuss the physical structure of the circuit and how everything ties together. In this chapter a practical

circuit-level implementation is proposed that uses GNRs for inner-gate connections and metal for gate to gate connection to reduce the total metal/graphene via count. This gate structure is simulated using the GNRFET model and the performance and power are compared to the 16nm CMOS technology. Specifically the contributions of this work are as follows:

- The first parameterized SPICE-compatible MOSFET type GNRFET model
- Modeling of graphene-specific variations, such as edge roughness, and width and T_{OX} variations
- Exploration of the design space of GNRFET for desirable transistor-level properties
- Insight into the use of GNR vs metal interconnects
- Comparison between the futuristic GNRFET circuits and traditional CMOS circuits

3.2 Building Circuits with GNRFETs

3.2.1 Properties and fabrication techniques

Graphene is a sheet of carbon atoms tightly packed into a two-dimensional honeycomb lattice. It is a zero band-gap material, which makes it an excellent conductor by nature [10]-[12]. Depending on the number of layers, graphene can be categorized into monolayer, bilayer, or multilayer graphene. Unlike CNTs, graphene does not wrap around and connect back to itself, so the unbounded edges are usually passivated by absorbents such as hydrogen, oxygen, hydroxyl group, carboxyl group, and ammonia [12]. Planar graphene must be processed into narrow strips (width < 10nm), known as GNRs, in order to open a band gap and turn it into a semiconductor [11], [12]. Theoretical work has shown that GNRs have energy gap inversely proportional to their widths [42], [43], [44]. In addition, edge states of GNRs define the energy gaps and determine the conductivity [42], [43]. GNRs with predominantly armchair edges are observed to be semiconducting, while GNRs with predominantly zig-zag edges demonstrate metallic properties. For this reason armchair GNRs are used to make transistors.

The GNR patterning can be accomplished with techniques such as lithography, chemical synthesis, or “unzipping” of carbon nanotubes [11], [12]. Due to the limitation of lithography resolution, the lithographic approach can only pattern GNRs down to ~20nm in width, and tends

to produce uneven edges, which undermine semiconductive properties [45], [46]. Chemical synthesis, on the other hand, is more controllable and is able to refine GNRs down to < 5nm in width. For example, in [45] ~ 2nm bilayer GNRs were fabricated and were shown to have an I_{ON}/I_{OFF} ratio as high as 10^5. These GNRs clearly demonstrate semiconducting property, which comes from band gap opening related to quantum confinement. However, the chemical synthesis process is less scalable, and thus is impractical for mass production [46]. Recently, a new method for producing narrow GNRs (~4nm) was proposed in [46], where the GNRs were patterned using standard lithography techniques and etching was used to narrow the GNRs from the edges. Further improvements in fabrication are necessary to realize large scale production of graphene circuits. There are also two varieties of GNRFETs: SB type and MOSFET type. SB type uses metal contacts and a graphene channel, which form a Schottky barrier at the junction In MOSFET type GNRFETs, the reservoirs (GNR not under the gate) are doped. Doping the reservoirs with donors will result in a NMOS like GNRFET, where current is dominated by electron conduction. If the reservoirs are doped with acceptors the GRNFET's current will be dominated by hole conduction and resemble a PMOS. According to numerical simulations [15], MOSFET type GNRFETs demonstrates a higher I_{ON}/I_{OFF} ratio and should outperform their SB type counterparts. That is the reason why this work focuses on modeling the MOSFET type GNRFET.

3.2.2 Proposed device structure

Figure 3.1 shows a proposed structure for a graphene transistor. This is similar to the structure proposed in [19], but the contacts are graphene instead of metal. Notice that multiple parallel ribbons are connected in parallel to increase drive strength and to form contacts of reasonable size. Each GNR is intrinsic under the gate (referred to as the channel) and is doped between the gate and the wide contact (referred to as the reservoir). Since the reservoirs are doped they are always conducting, while the channel is turned on and off with the gate. In Figure 3.1 and for the rest of this chapter L_{CH} is channel length, L_{RES} is the reservoir length, W_{CH} is the channel width of the actual ribbon, W_{gate} is the width of the entire gate, and SP is the spacing between the ribbons. Both the source and drain are also made of doped graphene.

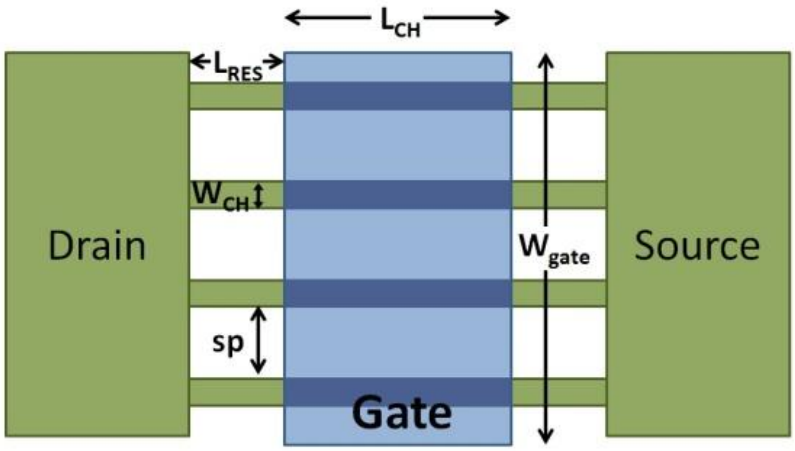

Figure 3.1 : Example layout of GNRFET

3.2.3 Circuit-level considerations

Since the contacts are made from graphene connection between the S/D of one transistor to the S/D of another transistor, such an inner gate connection could be made directly on the graphene layer. The width of the interconnect is set to equal the contact width. At this width both zig-zag and armchair GNR can serve as good conductors. We believe that it is optimal to have inner gate connections on graphene and to have gate to gate connections on conventional metal such as copper. Effectively the transistors and inner gate connections could be carved out from the same planar graphene sheet. This is demonstrated in Figure 3.2 by showcasing the connections for a NAND gate. Graphene connections are shown in thin red lines, metal connections are shown with thick blue lines, and the metal/graphene vias are shown as purple boxes.

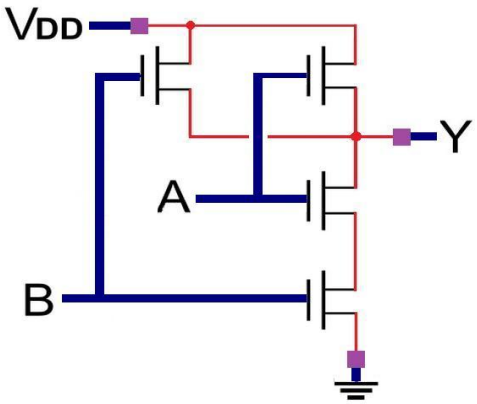

Figure 3.2: Use of metal vs. GNR connections in an NAND2 gate

Some works have also explored the use of graphene interconnects for longer gate to gate connections. In the performance of GNR interconnects were estimated and compared against copper. Graphene exhibits great conduction when there are a few layers, but if too many layers are stacked together it turns into graphite degrading the performance [41]. One solution is to use multi-layer GNRs that are doped with AsF_5, but even then the GNRs only outperform Cu under very specular edges, which are very difficult to fabricate. Nonetheless, this approach would also require the use of an additional graphene layer dedicated to interconnects. Note that the via resistance could easily dominate the wire delay. For example if it is made out of metal that would imply that it consists of two graphene metal junctions, both of which would have a large resistance [50]. One could conceive using a graphene via, but this would imply that electrons flow perpendicular to the graphene sheets and it is unclear what the effective resistance of such a configuration would be. Fabricating vertical graphene vias would be challenging as well. For all of these reasons, we believe that using metal gate-to-gate connections is more practical.

3.3 GNRFET Model

This section covers the modeling of GNRFET circuits. First, the single GNR ribbon is analyzed. Then its model is used to build a full transistor shown in Figure 3.3. Finally the modeling of the vias and graphene interconnects is also discussed. As will be shown in the experimental results, this model matches the simulation closely. In addition we propose

expressions for modeling the reservoir charge which is unique for MOSFET type GNRFETs (not present in the SB GNRFETs modeled in [18]). Details are presented below.

3.3.1 Single GNR model

Figure 3.3 depicts the equivalent SPICE circuit that can be used to simulate a single ribbon of GNRFETs. I_{DS} models the current flowing through the channel, while the capacitors $C_{CH,D}$, $C_{CH,S}$, $C_{G,CH}$, and $C_{SUB,CH}$ along with the voltage controlled voltage source V_{CH} are included to model the transient currents that result when the channel charges and discharges. V_{CH} is set to equal the potential of the channel (Ψ_{CH}), which will be discussed in greater detail in the following sections. $C_{G,CH}$ and $C_{SUB,CH}$ are the physical capacitors that model the coupling between the gate and the channel and the channel and the substrate respectively. $C_{CH,D}$ and $C_{CH,S}$ are effective capacitors that model the drain and source current that charges and discharges the channel. The expressions for computing these current sources and capacitors will be shown in the following subsections.

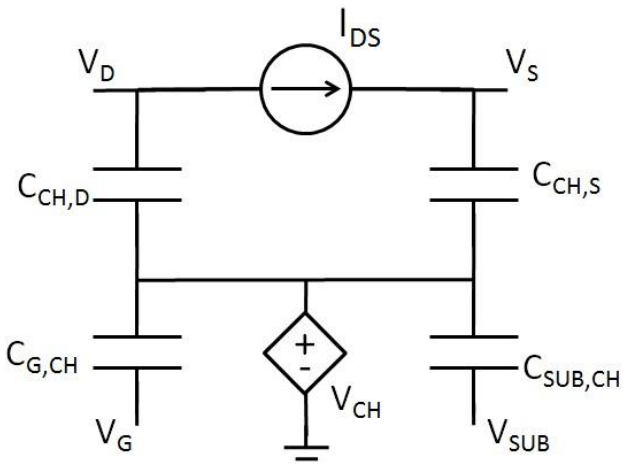

Figure 3.3: SPICE circuit modeling GNRFET

3.3.1.1 Computing the subbands

The positive subbands ε_α's can be computed based on [18] as shown in Equation 3.1. N (integer from 1 to N) is the number of dimmer lines, t is the tight-binding hopping parameter and

equals 2.7eV, α is the subband index, and $\delta\varepsilon_\alpha$ is an edge correction factor. Note that the negative subbands are symmetric and are simply the negative value of the computed ε_α's. Equation 3.2 can be used to compute the correction factor $\delta\varepsilon_\alpha$.

$$\varepsilon_\alpha = | t \times (1 + 2\cos(\frac{\pi\alpha}{N+1}) + \delta\varepsilon_\alpha | \tag{3.1}$$

$$\delta\varepsilon_\alpha = \frac{4vt}{N+1}\sin^2(\frac{\pi\alpha}{N+1}) \tag{3.2}$$

Note that the lowest lying subbands dominate the electrostatic and conduction properties. To improve the computation efficiency of the model, only the first two subbands are used. One can compute all of the subbands and then sort, but this would add to the computation time. First, α_0, which corresponds to a value of α that would make ε_α =0 (ignoring the correction factor) is computed with Equation 3.3. Now α_1 and α_2 just correspond to the two closest integer values of α_0 and can be found with Equation 3.4.

$$\alpha_0 = \frac{\cos^{-1}(-0.5) \times (N+1)}{\pi} = \frac{2N+2}{3} \tag{3.3}$$

$$\alpha_1 = floor(\alpha_0); \quad \alpha_2 = \alpha_1 + 1 \tag{3.4}$$

3.3.1.2 Finding Ψ_{CH}

The value of Ψ_{CH} is determined by the electrostatics in the channel. First the relationship between channel charge (Q_{CH}) and Ψ_{CH} will be derived. The value of Ψ_{CH} is determined by the electrostatics in the channel. In essence, the negative of the channel charge (Q_{CH}) has to equal the charge across the different capacitors that couple into the channel (Q_{CAP}). If both of these charges are expressed as a function of Ψ_{CH}, an equation solver (Figure 3.4) can be constructed in

SPICE to solve for the value of Ψ_{CH}. Thus the problem boils down to deriving expressions for Q_{CH} and Q_{CAP}.

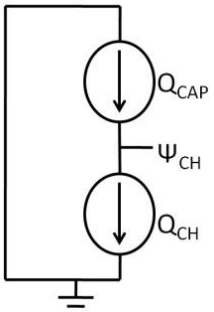

Figure 3.4: SPICE construct for solving Equation 3.22

3.3.1.2.1. Finding Q_{CH}

In order to find Q_{CH}, we need to know the concentration of the electrons and holes in the channel. In general the density of electrons in a given subband (ε_α) can be expressed as:

$$n_\alpha = \int_0^\infty f(\mathrm{E}) \times D_\alpha(E) dE \tag{3.5}$$

$$f(E) = \frac{1}{1 + \exp\left(\frac{\mathrm{E} - \mathrm{E_F}}{\mathrm{KT}}\right)} \tag{3.6}$$

$$D_\alpha(E) = \frac{2\sqrt{M_\alpha}}{\pi\hbar} \times \frac{(\varepsilon_\alpha + E)}{\sqrt{\varepsilon_\alpha E(E + 2\varepsilon_\alpha)}} \tag{3.7}$$

Here f(E) is the Fermi-Dirac distribution defined as $f(x) = (1+\exp(x/KT)^{-1}$, and $D_\alpha(E)$ is the density of states [18]. E is the energy and E_F is the Fermi level, both of which are referenced to the conduction band. K is the Boltzmann's constant, ħ is the reduced Planck's constant, and T is the temperature. M_α is the effective electron mass and can be found using Equation 3.8. Here a=2.46e-10m is the lattice constant.

$$M_\alpha = \frac{2\hbar^2 \varepsilon_\alpha}{3a^2t^2 \times \cos\left(\frac{\pi\alpha}{N+1}\right)} \tag{3.8}$$

There is no closed form solution for the integral in Equation 3.4, but the authors of [18] derive an expression by approximating f(E), by the Boltzmann's distribution f(E) = exp((E$_F$-E)/KT), which is valid when E-E$_F$> 3KT. As a result, their approximation is only valid when E$_F$ is below -3KT (which implies that its 3KT below conduction band). Since GNRs can have a low bandgap, this is not true for many of the bias conditions.

Thus we need to derive an expression that is valid across the entire range. Since Equation 3.4 cannot be solved directly, we approximate f(E) with an exponential function when E$_F$<E$_C$, a step function for (E$_F$-E$_C$>2KT), and a smoothing function is used in between. Since D$_\alpha$(E) is largest by the conduction band, it is most important to approximate f(E) accurately by the conduction band.

3.3.1.2.2. Exponential Approximation

For this method, f(E) is approximated by a decaying exponential function f'(E) as shown below.

$$f(E) \sim f'(E) = f(0) \times \exp\left(\frac{-E}{\beta \times KT}\right) \tag{3.9}$$

The term f($-E_{FC}$) is the fermi probability when E=E$_C$=0 and β is picked so that f(E$_{FC}$+3KT) = f'(E$_{FC}$+3KT) using Equation 3.10. This ensures that f'(E) approximates f(E) very well near the conduction band. Since this is where the density of states is highest, this provides an accurate estimate of n. The electron density computed with this approximation is denoted n$_{\alpha_exp}$ and can be found with Equation 3.11.

$$\beta(E_{FC}) = \frac{3}{\ln\left(f(-E_{FC}) \times \left(1 + \exp\left(\frac{3KT - E_{FC}}{KT}\right)\right)\right)} \tag{3.10}$$

$$n_{\alpha\_exp}(\mathrm{E_{FC}}) = \frac{\sqrt{M_\alpha \times (\beta KT)^3}(1 + \frac{2\varepsilon_\alpha}{\beta \mathrm{KT}})}{2\pi\hbar\varepsilon_\alpha} \times \exp(\frac{\mathrm{E_{FC}}}{\beta \mathrm{KT}}) \quad (3.11)$$

3.3.1.2.3. Step Approximation

Now consider the case when E_{FC} >3KT. In this situation the f(E) is close to 1 around the conduction band. Since the density of states is highest in this region, approximating the Fermi-Dirac distribution as a step function (1 for E < E_F and 0 for E > E_F) provides a good approximation for the electron density. As a side benefit, the Fermi-Dirac distribution is symmetric so the overestimation that occurs for E_F-E_C >0 is compensated by the under-estimation that occurs for E_F-E_C <0. The electron density computed with this assumption is denoted n_{α_exp} and can be computed with the expression below. Note that for E_{FC}<0 the expression evaluates to 0.

$$\begin{aligned} n_{\alpha\_step}(E_{FC}) &= \int_0^{\mathrm{E_{FC}}} \frac{2\sqrt{M_\alpha}}{\pi\hbar} \times \frac{(\varepsilon_\alpha + E)}{\sqrt{\varepsilon_\alpha E(E + 2\varepsilon_\alpha)}} dE \\ &= \frac{2\sqrt{M_\alpha}}{\pi\hbar} \times \sqrt{\max(\frac{E_{FC}(E_{FC} + 2\varepsilon_\alpha)}{\varepsilon_\alpha}, 0)} \end{aligned} \quad (3.12)$$

3.3.1.2.4. Combined Approximation

Two expressions have been derived, which approximate the electron density well under different conditions. To ensure a smooth continuous function for charge the n_α can be expressed as a weighted sum of the two approximations. The final result is expressed in Equation 3.13, where n_{α_exp} is the exponential approximation from Equation 3.11 and n_{α_step} is the step approximation from Equation 3.12, and m is the relative weight and is defined with Equation 3.14. Note that if E_{FC} =0, both of these approximations are weighted equally; if E_{FC} <0 the exponential approximation has the major contribution; and for E_{FC} > 2KT the step approximation has the major contribution.

$$n_\alpha(E_{FC}) = m \times n_{\alpha\_exp}(E_{FC}) + (\mathrm{m}-1) \times n_{\alpha\_step}(E_{FC}) \tag{3.13}$$

$$m = \frac{1}{1 + \exp\left(\frac{(\mathrm{E_{FC}} - \mathrm{KT}) \times 3}{\mathrm{KT}}\right)} \tag{3.14}$$

The effectiveness of this approximation was tested for ε_α = 0.1 – 0.5 and good accuracy was observed for all cases. An example for ε_α = 0.3 corresponding to N=12 is shown in Figure 3.5. Here the electron density is plotted against the potential difference between the Fermi level and the conduction band. The numerical result was taken by evaluating the integral in Equation 3.5, the combined result is based on Equation 3.13, the exponential result is based on Equation 3.11, and the Boltzmann result corresponds to Equation 3.8 based on expressions from [18]. The ε_α is set to 0.3eV, which corresponds to the N=12 case. One can see that at low bias all of the expressions are accurate. As expected, the Boltzmann approximation fails first, followed by the exponential approximation. The combined approximation is accurate throughout the entire range.

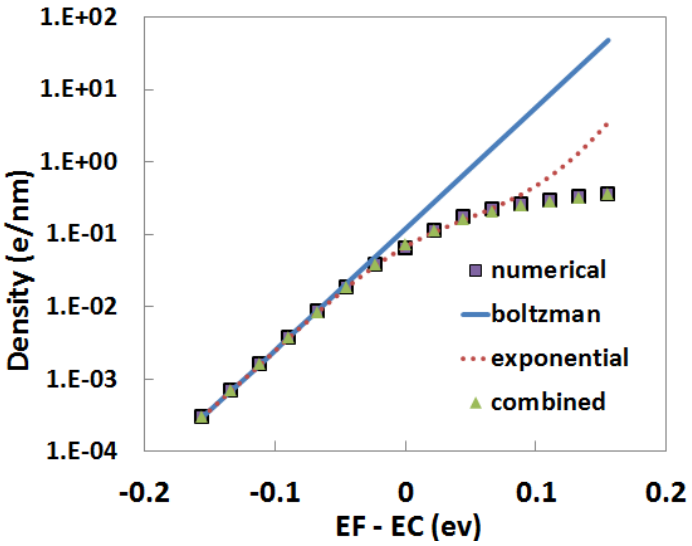

Figure 3.5: Comparisons of analytical approximations

3.3.1.2.5. Computing total channel charge

Similar analysis can be performed for the hole concentration to obtain Equations 3.15-3.19. They are essentially the same, except for the input parameter E_{VF}, which corresponds to the energy difference between the valence band and the Fermi-level.

$$\beta(E_{VF}) = \frac{3}{\ln(f(0) \times (1 + \exp(\frac{3KT - E_{VF}}{KT})} \tag{3.15}$$

$$h_{\alpha\_exp}(E_{VF}) = \frac{\sqrt{M_\alpha \times (\beta KT)^3}(1 + \frac{2\varepsilon_\alpha}{\beta KT})}{2\pi\hbar\varepsilon_\alpha} \times \exp(\frac{E_{VF}}{\beta KT}) \tag{3.16}$$

$$h_{\alpha\_step}(E_{VF}) = = \frac{2\sqrt{M_\alpha}}{\pi\hbar} \times \sqrt{\max(\frac{E_{VF}(E_{VF} + 2\varepsilon_\alpha)}{\varepsilon_\alpha}, 0)} \tag{3.17}$$

$$h_\alpha(E_{VF}) = m \times n_{\alpha\_exp}(E_{VF}) + (1 - m) \times n_{\alpha\_step}(E_{VF}) \tag{3.18}$$

$$m = \frac{1}{1 + \exp\left(\frac{(E_{VF} - KT) \times 3}{KT}\right)} \tag{3.19}$$

When analyzing Q_{CH}, it is helpful to look at the band-diagram shown in Figure 3.6. Here the GNRFET is biased with a positive V_{GS} and a positive $V_{DS.}$ E_{FS} and E_{FD} correspond to the Fermi level at the source and drain respectively. Since V_{DS} is positive, E_{FD} is below E_{FS}. Note that E_{FS} and E_{FD} are both above the conduction band of the source and drain, because they are heavily doped and have a large concentration of electrons. Note that at moderate bias there are no holes in the channel. However, if the conduction band on the drain side (E_{CD}) is below the valence band of the channel (E_{VCH}), electrons can tunnel from the drain through the band into the channel.

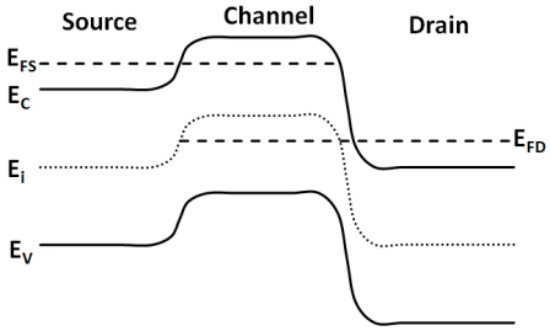

Figure 3.6: Example Band Diagram

This probability ($Tr(\Psi_{CH,D})$) is captured with Equation 3.20. Here $\Psi_{CH,D}$ is the amount of band bending between the channel and drain, $\eta_{0.5}$ is a fitting parameter, which captures the amount of band-bending beyond (E_G) necessary for transmission to equal 0.5. γ is another fitting parameter, which controls how fast Tr goes from zero to one as $\Psi_{CH,D}$ is increased. Note that $\eta_{0.5}$ and γ are the same for all the GNRFET device settings so they do not need to be recomputed for different devices.

$$Tr(\Psi_{CH,D}) = [1 + \exp\left(\frac{(2+\eta_{0.5})\varepsilon_\alpha - \Psi_{CH,D}}{\gamma \times \varepsilon_\alpha}\right)]^{-1} \tag{3.20}$$

The final expression for channel charge is shown in Equation 3.21. L_{CH} is the channel length, q is the electron charge, n_α is the electron density obtained from Equation 3.13, and h_α is the hole density obtained from Equation3.18. Note that the channel potential (Ψ_{CH}) is the negative of the intrinsic energy level (E_i) and thus the conduction band $E_C=\varepsilon_\alpha - \Psi_{CH}$ and the valence band $E_V = -\varepsilon_\alpha - \Psi_{CH}$.

$$Q_{CH}(\Psi_{CH}, V_D, V_S) = \frac{qL_{CH}}{2}\sum_\alpha [-n_\alpha(\Psi_{CH} - \varepsilon_\alpha - V_S)] - \tag{3.21}$$

$$-n_\alpha(\Psi_{CH} - \varepsilon_\alpha - V_D) + Tr(V_{bi,chd}) \times h_\alpha(V_D - \Psi_{CH} - \varepsilon_\alpha)]$$

3.3.1.2.6. Computing Q_{CAP}

Q_{CAP} can be found with Equation 3.22. $C_{G,CH}$ is the capacitance between the gate and the channel, $C_{SUB,CH}$ is the capacitance between the substrate and the channel, and V_{FB} is the flat-band voltage, which is the work function difference between the metal and graphene. C_1 and C_2 are fitting parameters used to model the drain-induced barrier lowering effect. Note that in a typical GNRFET that has multiple ribbons in parallel, the gate width (W_{gate}) will be larger than the width of the GNR (W_{CH}) and the oxide thickness (T_{ox}). As a result the expression for the micro-strip gives a fair approximation. Since W_{gate} is not infinite, a correction term $(1+1.5T_{ox}/W_{gate})^{-1}$ was added resulting in Equation 3.23. Note that for $T_{ox}<<W_{gate}$ this term goes to 1, simplifying the expression back to the micro-strip form. This expression was verified for accuracy against a numerical electro-magnetic field solver [47]. This equation can also be used to compute $C_{SUB,\,CH}$ by setting T_{OX} to the substrate thickness, and W_{gate} to substrate width, which can be assumed to be infinity

$$Q_{CAP} = C_{G,CH}(V_G - V_{FB} - \Psi_{CH}) + C_{SUB,CH}(V_{SUB} - V_{FB} - \Psi_{CH}) + C_1(V_D - \Psi_{CH}) + C_2(V_S - \Psi_{CH}) \quad (3.22)$$

$$C_{G,CH} = \frac{L \times 5.55 \times 10^{-11} \times \varepsilon_R}{\left(1 + 1.5 \times \frac{T_{ox}}{W_{gate}}\right) \times \ln\left(\frac{5.98 \times W_{ch}}{0.8 \times T_{ox}}\right)} \quad (3.23)$$

3.3.1.3 Computing the intrinsic capacitors

For the model in Figure 3.3 to be functional all of the capacitors have to be computed. $C_{G,CH}$ and $C_{SUB,CH}$ can be computed using Equation 3.23 as shown in the previous section. $C_{CH,D}$ and $C_{CH,S}$ are derivatives of the channel charge with respect to drain and source respectively. Since the charge expression is complex, taking its derivative analytically is complicated. It is much simpler to take it numerically as shown in Equation 3.27. The size of ΔV_D is arbitrary and is set to 0.1mV in the presented model. The same procedure can be performed to compute $C_{CH,S}$.

$$C_{CH,D} = \frac{Q_{CH}(\Psi_{CH}, V_D + \Delta V_D) - Q_{CH}(\Psi_{CH}, V_D)}{\Delta V_D} \tag{3.24}$$

3.3.1.4 Current modeling

If the Ψ_{CH} is known, the electron current (I_e) can be computed using the Landauer-Buttiker formalism [18], [48] as shown in Equation 3.25, where q is the electron charge, h is Planck's constant, T(E) is the transmission probability, f is the Fermi probability defined in the same way as Equation 3.5. $E_{FS,C}$ and $E_{FD,C}$ are difference between the energy level of the conduction band and the source-referred and drain-referred Fermi level, respectively. Essentially, the probability of the electrons being injected onto the conduction band from the source side is subtracted from the probability of the electrons being injected onto the conduction band from the drain side.

$$I_e = \frac{2q}{h} \sum_{\alpha} \int_0^{\infty} T(E)[f(E - E_{FS,C}) - f(E - E_{FD,C})]\, dE \tag{3.25}$$

When developing the compact model, the transmission coefficient for electrons is assumed to be 1, since the reservoirs are doped N-type and the hole transmission probability is assumed to be 0. These are the same assumptions made in [49], when developing the Stanford CNT compact model. Here the thermionic current is computed by subtracting the electrons injected into the drain from the electrons being injected into the source. With this assumption, recognizing the Fermi-Dirac integral of order 0, the integral in Equation 3.25 can be evaluated analytically resulting in Equation 3.26.

$$\begin{aligned} I_e(\Psi_{CH}, V_D, V_S) = \frac{2q}{h} \sum_{\alpha} \ln\left(1 + \exp\left(\frac{q(\Psi_{CH} - V_S) - \varepsilon_\alpha}{KT}\right)\right) - \\ -\ln\left(1 + \exp\left(\frac{q(\Psi_{CH} - V_D) - \varepsilon_\alpha}{KT}\right)\right) \end{aligned} \tag{3.26}$$

3.3.2 Full GNRFET model

As shown in Figure 3.7, a complete GNRFET consists of multiple parallel ribbons. The gate source and drain will be shared among all of these independent ribbons. Figure 3.7 shows a SPICE-level implementation of the full GNRFET. The transistors highlighted in red correspond to the individual GNRs modeled by the circuit in Figure 3.3. In addition, one must add the parasitic capacitors $C_{g,s}$ and $C_{g,d}$, which are caused by the fringing fields between the gate and the reservoirs. There is no direct analytical expression for computing these. Thus an electro-magnetic field solver was used to compute the capacitance under various transistor settings and an empirical equation was derived to match the data. Equation 3.27 is used for the model. Both T_{OX} and W_{GATE} should be entered in nanometers.

$$C_{gd} = C_{gs} = 1.26 \times 10^{-19} F/nm \times W_{gate} \times (0.8nm - 0.2 \times T_{ox} + 0.015/nm \times {T_{ox}}^2) \quad (3.27)$$

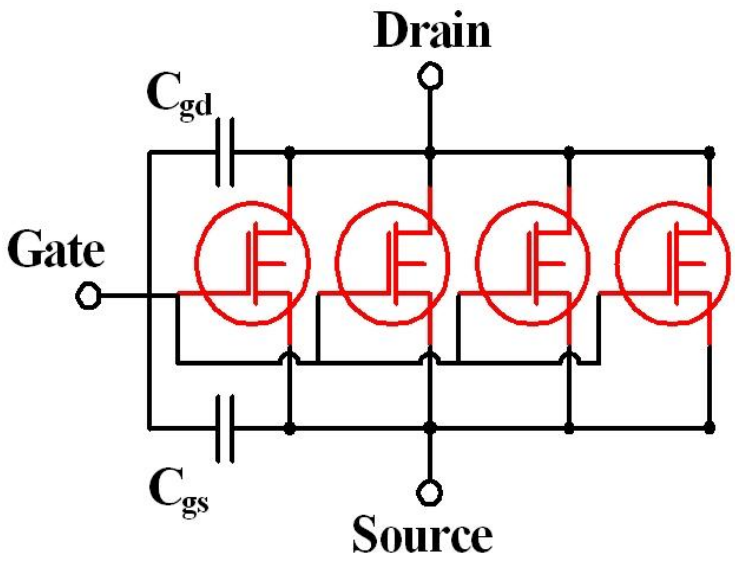

Figure 3.7: SPICE model for full GNRFET

3.3.3 Modeling vias and interconnects

The graphene interconnects between transistors are quite short and are much less than the mean free path of graphene (1μm) and should have negligible resistance. For this reason a first order model can neglect the resistance of these inner gate interconnects (same assumption was

made in [19]). On the other hand the impact of the metal/graphene connection is quite significant. This contact resistance can be estimated based on previous experimental results from [50]. In this work the graphene/metal resistance was measured for various metals using a 4-point measurement. In this approach, two outside terminals are used to supply a current across a load, while two other terminals are used to measure the voltage. With this method the resistance of the measuring probes does not add to the calculated value. It was found that the resistance is proportional to the contact width. Nickel was the best with a contact resistance of 1kΩ× μm (1 μm width connection will have resistance of 1kΩ). Other metals such as Ti/Au and Cr/Au can also be used, but their resistance was much worse (5kΩ/μm - 100kΩ/μm depending on setting).

3.4 Model Verification

In this section, the compact model is verified against Nano TCAD ViDES [14], which is a publically available numerical device simulator. It is based on an atomistic tight binding Hamiltonian, non-equilibrium Green's functions formalism, and three-dimensional electrostatics. Transport is assumed to be fully ballistic. As described in the previous section the model utilizes a few fitting parameters. C_1 and C_2, which help model the drain induced barrier lowering effect, were set to $0.15C_{G,CH}$×Tr and $0.05C_{G,CH}$ respectively. $\eta_{0.5}$ was set to 0.6 and γ was set to 1/6.

The IV comparison is shown in Figure 3.8. The current is plotted against V_{GS} for V_{DS}=0.1V and V_{DD}=0.5V. Overall, the model matches the numerical results quite well. This device had T_{OX}=1nm, N=12, L=15nm, doping fraction = 0.005, and reservoir length of 10nm.

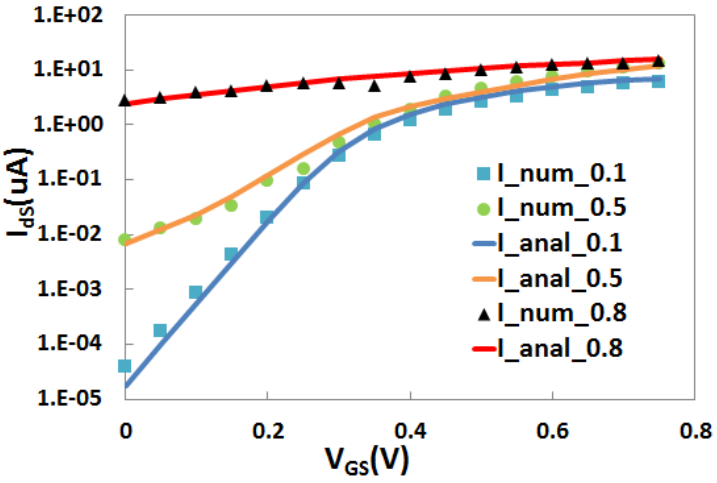

Figure 3.8 : IV comparison for GNRFET

To ensure the accuracy of the compact model, it is tested at various device settings. The model is compared to Vides at different values of length, oxide thickness, number of dimmer lines, and doping. During this comparison, the focus is on I_{ON} (current when $V_{GS}=V_{DS}=V_{DD}$) and I_{OFF} ($V_{GS}=0$, $V_{DS}=V_{DD}$), because they are a good indicator of performance and leakage power. For all of these tests $V_{DD}=0.5V$, because that is the expected nominal operating supply voltage. First, the I_{ON} and I_{OFF} results from Vides and our compact model are compared for various numbers of dimmer lines, which is shown in Figure 3.9. Since Vides only supports even number of N, these were the only ones shown. Nevertheless, it is clear that the model tracks the periodic effect of the bandgap very well. For N=8 and N=14, the bandgap is very small, which results in a very poor I_{ON}/I_{OFF} ratio. For N=6 and N=12, there is a moderate bandgap, which results in good I_{ON}/I_{OFF} ratio and a high I_{ON}. For N=16 and N=10, the bandgap is largest which results in the highest I_{ON}/I_{OFF} ratio. However, the I_{ON} is still fairly low, which can hinder the propagation delay.

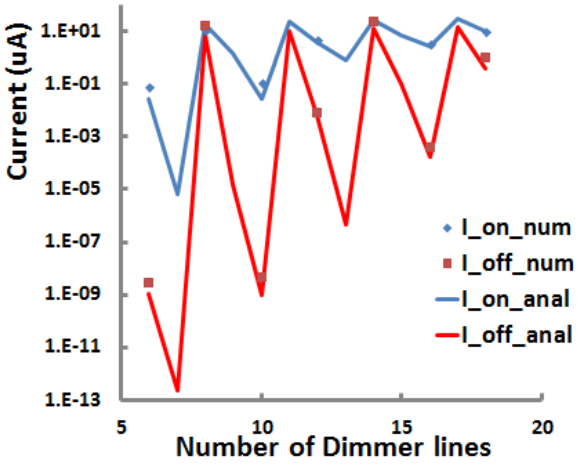

Figure 3.9: Current vs. number of dimmer lines

Next the effect of oxide thickness is examined. Figure 3.10 compares the results from the compact model with vides. T_{OX} is inversely related to $C_{G,CH}$. Thus a smaller T_{OX} implies a larger C_{OX}, which would result in better controller of the channel potential. Thus I_{ON} is increased and I_{OFF} is reduced. The compact model models this effect well and matches well with results from vides. Figure 3.11 verifies the model for doping. Doping affects the band-bending between the channel and the source ($\Psi_{CH,D}$). As a result, higher doping leads to a higher Tr. We expect T_R to be zero at low doping levels and to be one at high doping levels (under correct bias). From Figure 3.11, we can infer that for the I_{OFF} bias, T_R is zero for doping levels of 0.0005 and 0.001 and is one for doping above 0.005. Note that the current in our model is independent of length. Based on results from ViDES, this held true for L_{CH} between 15nm and 30nm. However, at L_{CH} <10nm, I_{OFF} started to increase. Most likely at a very small L_{CH}, the contribution of electrons diffused from the reservoirs becomes non-negligible.

Finally, the compact modeling of the edge roughness (values on the x axis corresponds to the probability that any outside atom is removed) is verified against ViDES as shown in Figure 3.12. As mentioned earlier, edge roughness tends to reduce the on current and also reduces the bandgap, which leads to an increase in the off current. Even though our model does not match the ViDES data perfectly, it clearly captures the deterioration of the device performance as the edge roughness is increased. Thus it will still paint a clear picture of how edge roughness affects circuit level performance.

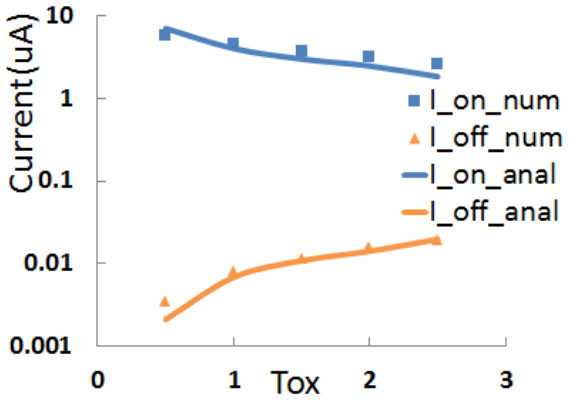

Figure 3.10 : Current vs. oxide thickness

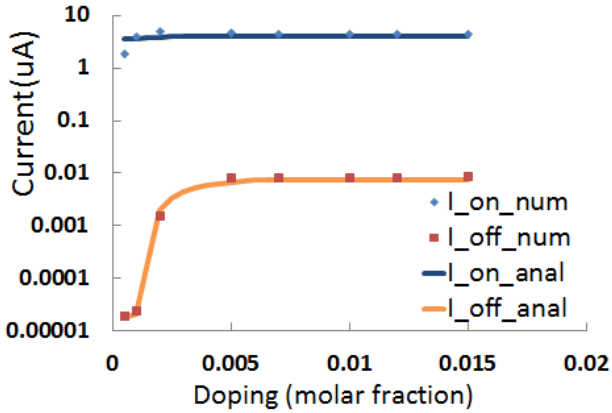

Figure 3.11: Current vs. doping

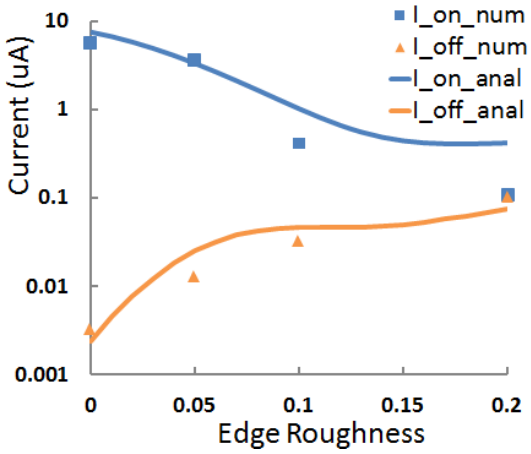

Figure 3.12: Current vs. roughness

3.5 Gate-Level Evaluation

With the validity and accuracy of our SPICE model thoroughly verified, we can proceed with circuit-level evaluation of GNRFETs. This gives insightful information on how graphene-based circuits would perform once the fabrication techniques become mature. We first evaluated the delay and power of a buffer chain under various supply voltages to understand the power delay trade-off. Then the GNRFET buffer chain is simulated under various structural parameters such as supply voltage, doping, oxide thickness, and length, to evaluate the impact of process variations. Finally, a thorough comparison is performed on a set of basic digital circuits built on GNRFET and CMOS under their respective optimal settings. For these SPICE simulations, the input slew of 10ps was used, and a 1fF load was added to the outputs.

3.5.1 Impact of supply voltage

We performed tests on a 7-stage, fanout-of-4 buffer chain in CMOS, ideal GNRFETs (without metal-graphene contact resistance), GNRFETs with metal-graphene contact resistance, and GNRFETs with metal-graphene contacts and edge-roughness. The CMOS was implemented with the 16nm High-Performance library from Predictive Technology Model (PTM [51]), and GNRFETs were implemented with our SPICE model. The minimum size GNRFET is set to have six ribbons so that the total gate width equals 32nm just like CMOS. GNRFET metal-graphene junctions are present in circuit layouts as discussed in section 3.2.3 and are modeled with 20kΩ resistors (assuming 50nm via width) at all metal/graphene junctions (example for NAND2 shown in Figure 3.2). Limitations on fabrication techniques contribute to the inevitable imperfection of GNR's ideal smooth edges, which is edge-roughness. The effect of edge-roughness can be simulated in our SPICE model by setting the percentage of roughness (p). In our case, we simulated p=5% and 10%. Considering graphene-metal contacts and edge-roughness makes our simulations closer to reality. The ideal GNRFET, although not practical, gives an upper bound on GNRFET circuit performance under ideal conditions.

Figure 3.13 shows the delay, dynamic, and leakage power of the buffer chain under V_{DD}=0.3 to 0.8V. For GNRFETs, the delay reduced tremendously from V_{DD} =0.3V to 0.5V. As

expected, dynamic power gradually grows with V_{DD}. Leakage power grows significantly with V_{DD} since higher drain voltage gives much higher I_{OFF}. It can be observed that the optimal operating V_{DD} is around 0.5V, if all three metrics are considered. Compared to the ideal GNRFET, contact resistance and edge-roughness greatly deteriorate the performance. CMOS performs fairly well in the range of V_{DD} =0.6 to 0.8V. At V_{DD} =0.3V, the CMOS circuit does not operate correctly so we omitted the results. It is worth noting that even non-ideal GNR circuits can outperform CMOS in delay when V_{DD} < 0.6V. Note that the effective area under the gate is larger for CMOS, because there is spacing between GNRs. This results in higher input capacitance and hence larger dynamic power.

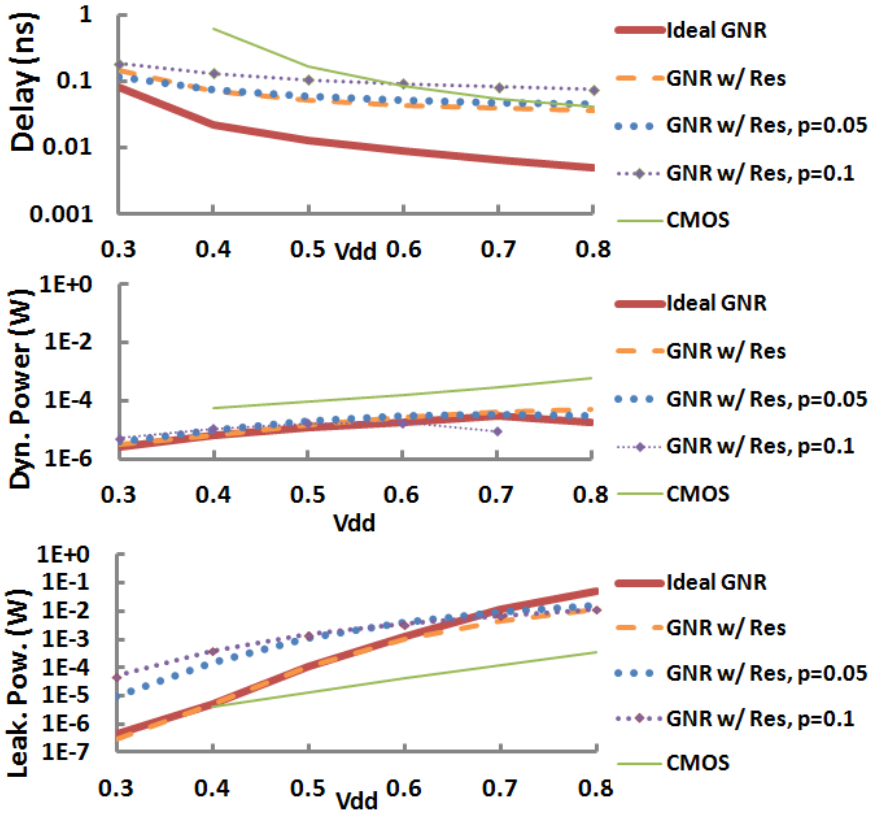

Figure 3.13: Delay and power vs. supply voltage

3.5.2 Impact of process variations

Process variation on GNRFETs will result in fluctuations in channel width (W_{CH}), length (L_{CH}), oxide thickness (T_{OX}), and doping level. With our SPICE-compatible compact model, we are able to evaluate the impacts on circuit performance due to these variations. Width directly determines the number of dimmer lines, N, of the GNR structure, and N has a great impact on the subbands, and hence the IV characteristics as was shown in section 4.1. As a result, width variation in GNRFET can contribute to significant changes in circuit performance. Note that width variation is not to be confused with edge-roughness, which describes the smoothness (roughness) of the edges of GNR instead of the effective width of the GNR.

We simulated the buffer chain presented in the previous section under different width settings (represented by N), again, for ideal GNRFETS, GNRFETs with contact resistance, and GNRFETs with contact resistance and edge-roughness. Figure 3.14 shows the results, which are consistent with Figure 3.9. N=10 gives high delay and low power due to its low I_{ON} and I_{OFF} currents. N=8 and 14 have almost equally high I_{ON} and I_{OFF}, and thus the delay is low while the leakage power is extremely high. When edge-roughness is present, the effective N falls between N and N-2, and the effective subbands fall between those of N and N-2 as well. Moreover, GNRFETs with higher edge-roughness tend to be affected less by the periodic behavior. This explains the dramatic difference between the ideal GNR and GNR with p=0.1 at N=8 and 14 since the Ion/Ioff ratios at N=7 and 13 are both high.

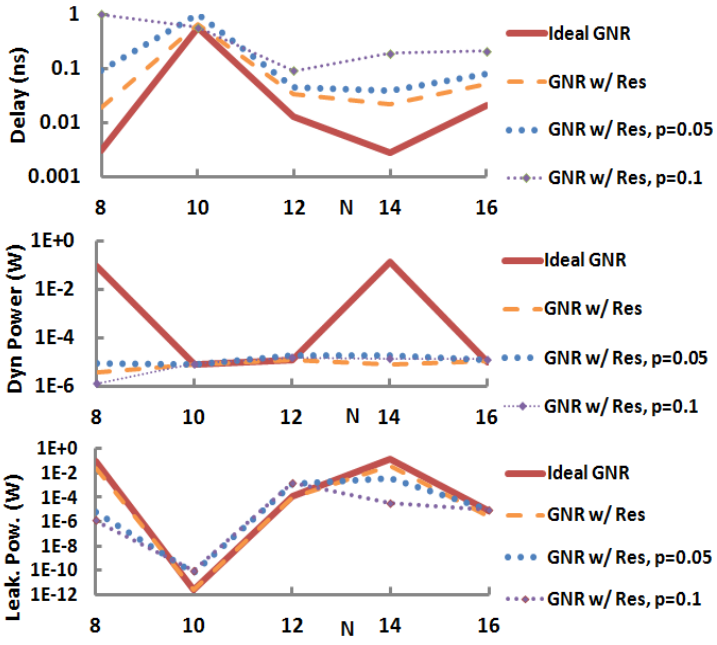

Figure 3.14: Delay and power vs. N

To evaluate the impact of variation on other structural parameters, we performed a series of SPICE simulations by varying L_{CH}, T_{OX}, and doping to see their respective impacts at the circuit level. Again, we simulated ideal GNRFETs, GNRFETs with contact resistance, and GNRFETs with contact resistance and edge-roughness. Table 3.1 shows the results. We first picked a nominal setting of L_{CH} =15nm, T_{OX} =1nm, and doping=0.005 as in 3.4.1. Then, each of L_{CH}, T_{OX}, and doping was varied one at a time between their respective max and min values. We reported the max and min delay, dynamic power, and leakage power with respect to the variable in concern to show the impact these structural changes can have on the circuit. These results are presented in Table 3.1. Among L_{CH}, T_{OX}, and doping, L has least effect, T_{OX} has an impact on everything, and doping greatly changes the leakage power. As discussed in 3.4.1, gate input capacitance is related to L_{CH} and T_{OX}, I_{ON} is affected by T_{OX} as well, and doping controls I_{OFF}. I_{ON} and input capacitance affects the delay, while I_{OFF} contributes to leakage power, which is consistent with the results in Table 3.1.

Table 3.1: Effect of Variations on GNRFET Circuits

		Parameters		Delay (ns)		Dynamic Power (W)		Leakage Power (W)	
		Max	Min	Max	Min	Max	Min	Max	Min
Ideal GNR	L	25nm	10nm	0.0155	0.0114	1.226E-05	1.071E-05	1.172E-04	1.103E-04
	Tox	2.5nm	0.5nm	0.0204	0.0094	1.386E-05	9.250E-06	2.741E-04	3.522E-05
	Doping	0.015	0.001	0.0137	0.0118	2.426E-05	1.022E-05	1.253E-04	6.680E-07
GNR w/ Res	L	25nm	10nm	0.0407	0.0300	1.515E-05	1.169E-05	1.087E-04	1.086E-04
	Tox	2.5nm	0.5nm	0.0399	0.0335	1.549E-05	1.015E-05	2.586E-04	3.379E-05
	Doping	0.015	0.001	0.0369	0.0306	1.224E-05	1.018E-05	1.229E-04	2.096E-07
GNR w/ Res, p=0.05	L	25nm	10nm	0.0549	0.0379	2.288E-05	1.563E-05	1.270E-03	1.258E-03
	Tox	2.5nm	0.5nm	0.0514	0.0433	2.643E-05	3.750E-06	2.607E-03	3.454E-04
	Doping	0.015	0.001	0.0449	0.0414	1.775E-05	1.333E-05	1.282E-03	2.881E-05
GNR w/ Res, p=0.10	L	25nm	10nm	0.1136	0.0770	1.950E-05	1.363E-05	1.616E-03	1.598E-03
	Tox	2.5nm	0.5nm	0.1132	0.0837	3.211E-05	4.375E-06	2.240E-03	6.573E-04
	Doping	0.015	0.001	0.0906	0.0895	1.989E-05	1.413E-05	1.611E-03	1.676E-04

3.5.3 Performance comparison against CMOS

Here we present a case study comparing the delay and power performance on a set of digital circuits, implemented with CMOS and GNRFETs, respectively. We choose the CMOS technology node to be the 16nm HP library from PTM with nominal V_{DD}=0.7V, which is the default for this library. According to the exploration in 3.5.1 and 3.5.2, GNRFETs with N=12, doping=0.001, and V_{DD}=0.5V are predicted to have a low delay and power product, and hence we choose to adopt this setting in our circuit simulations. To match the dimensions of CMOS, our GNRFET is set to have six parallel nano-ribbons, L_{CH}=16nm, and T_{OX}=0.95nm. Again, we also implemented the circuits in GNRFETs with metal-graphene contact and edge-roughness in order to model the reality more closely. The set of circuits we simulated include an inverter, 2-input nand and nor gates, 3-input nand and nor gates, the buffer chain, and c17 from the ISCAS85 benchmarks. Table 3.2 shows the results. It can be observed that GNRFETs, even with practical imperfections, can outperform CMOS in terms of power consumption. CMOS performs better in delay unless the GNR is ideal. Specifically, comparing to CMOS for performance, GNRFET *ideal*, *w/ res*, *w/ res + p=0.05*, *w/ res + p=0.1* are 46.1% faster, 11.3% slower, 61.2% slower, and 288.4% slower on average, respectively. This shows that edge roughness plays a significant role degrading ballistic transport in the GNRFET. Comparing to CMOS for dynamic power, GNRFET cases are 72.4%, 72.4%, 71%, and 65% lower respectively, due to lower V_{DD} and lower GNRFET gate capacitance. For leakage power, they are 99.6%, 99.6%, 78.1%, and

Table 3.2: Comparison to CMOS

Circuit	Delay				
	Ideal GNR	GNR w/ Res	GNR w/ Res, p=0.05	GNR w/ Res, p=0.1	CMOS
inv	1.28E-02	2.68E-02	3.77E-02	7.93E-02	1.45E-02
nand2	1.74E-02	3.50E-02	5.30E-02	1.22E-01	3.70E-02
nor2	1.91E-02	3.45E-02	5.31E-02	1.24E-01	2.00E-02
nand3	2.30E-02	4.40E-02	7.00E-02	1.70E-01	7.10E-02
nor3	2.50E-02	4.30E-02	6.63E-02	1.70E-01	4.00E-02
ring osc	1.16E-02	3.08E-02	4.19E-02	1.79E-01	5.50E-02
c17	1.83E-02	6.09E-02	7.16E-02	1.40E-01	6.00E-02
Circuit	Dynamic Power				
	Ideal GNR	GNR w/ Res	GNR w/ Res, p=0.05	GNR w/ Res, p=0.1	CMOS
inv	3.37E-08	3.38E-08	3.34E-08	3.06E-08	9.87E-08
nand2	4.10E-08	4.07E-08	4.04E-08	3.67E-08	1.22E-07
nor2	4.06E-08	4.07E-08	4.02E-08	3.65E-08	1.20E-07
nand3	2.72E-08	2.67E-08	2.63E-08	2.30E-08	8.74E-08
nor3	2.66E-08	2.65E-08	2.61E-08	2.28E-08	9.03E-08
ring osc	1.01E-05	1.04E-05	1.36E-05	1.37E-04	2.88E-04
c17	5.85E-08	5.85E-08	8.19E-08	1.16E-07	2.10E-07
Circuit	Leakage Power				
	Ideal GNR	GNR w/ Res	GNR w/ Res, p=0.05	GNR w/ Res, p=0.1	CMOS
inv	6.12E-11	6.12E-11	4.44E-09	2.72E-08	2.10E-08
nand2	2.97E-11	2.97E-11	3.72E-09	1.90E-08	2.00E-08
nor2	1.22E-10	1.22E-10	3.72E-09	1.90E-08	1.39E-08
nand3	5.38E-11	4.55E-11	2.58E-09	1.12E-08	1.44E-08
nor3	3.80E-11	4.04E-11	2.58E-09	1.12E-08	8.77E-09
ring osc	3.34E-07	3.34E-07	2.63E-05	1.33E-04	1.22E-04
c17	9.40E-10	1.10E-09	7.82E-08	4.01E-07	4.35E-07

9.6% lower respectively, due to low off current when doping level is low. However, the I_{ON}/I_{OFF} ratio quickly deteriorates with a p=0.1 edge roughness (Figure 3.12).

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

In this thesis, the impact of process variations on circuit performance was analyzed. It was shown that considering process variations is critical during the design of circuits or the evaluation of a new technology. As the semiconductor industry continues to scale down the critical dimension, the study of these effects will become more challenging and more important.

4.1 Temperature-Aware SSTA

In this chapter a new SSTA method was demonstrated that accounts for the statistical thermal profile of the circuit and closely matches the MC simulation results. It was also shown that previous approaches that assume deterministic temperature lead to inaccurate mean and SD estimations. However, we found that such an approach can still give reasonable estimates for high yield targets. Thus, if run-time is a concern, one can obtain a decent yield estimate by assuming nominal leakage power. On the other hand, if one targets lower yield (<85%) and desires higher accuracy, it is crucial to account for the variability introduced by the leakage power. Note that we assumed the activity factor to be constant over time. For a real chip, this may not be true. However, one can run the tool after setting the constant activity factor to the worst case. This would give the designer the timing yield under worst-case dynamic power. Finally, considering the impact of PVs on dynamic power can be a future work.

4.2 Compact GNRFET Model for Technology Exploration

A SPICE-compatible compact model for a MOSFET-type GNRFET was presented, which will be released as open source in the future. It captured the effect of N, T_{OX}, edge roughness, doping, and length on the device characteristics. This model was used to perform circuit level evaluation for GNRFET-based circuits. In general we found that circuits that used optimized GNRFET structure and ideal metal/graphene junction and smooth edges can significantly outperform the CMOS counterparts, and GNRFET power is consistently better than

its CMOS counterpart across the board, except for the leakage power when edge roughness is 10%. However, GNRFETs are extremely sensitive to the number of dimmer lines and edge roughness. Thus the future design of complex reliable circuits will require fabrication techniques that offer precise control of the graphene material.

REFERENCES

[1] International Technology Roadmap for Semiconductors. (2010, November) [Online]. http://www.itrs.net/

[2] A. Agarwal et al., "Statistical delay computation considering spatial correlations," in *IEEE/ACM ASP DAC*, 2003, pp. 271-276.

[3] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *ACM/IEEE ICCAD*, 2003.

[4] M. Orshansky and A. Bandyopadhyay, "Fast statistical timing analysis handling arbitrary delay correlations," in *IEEE/ACM DAC*, 2004.

[5] NANGATE. (2010, July) [Online]. www.nangate.com

[6] Predictive technology model. (2010, July) [Online]. http://ptm.asu.edu/

[7] V. Kursun and R. Kumar, "Impact of Temperature Fluctuations on Circuit Characteristics in 180nm and 65nm CMOS Technologies ," in *IEEE ISCS*, 2006.

[8] B. Lasbouygues, R. Wilson, and N. Maurine, "Temperature- and Voltage-Aware Timing Analysis," *IEEE TCAD*, vol. 26, no. 4, pp. 801-815, March 2007.

[9] L. Peng, "Critical Path Analysis Considering Temperature, Power Supply Variations and Temperature Induced Leakage," in *ISQED*, 2006, pp. 254-259.

[10] A. K. Geim and K. S. Novoselov, "The Rise of Graphene" *Nature Materials*, vol. 6, pp. 183-191, March, 2007.

[11] S. Chilstedt, C. Dong, and D. Chen, *Carbon Nanomaterial Transistors and Circuits, Transistors: Types, Materials, and Applications*, Nova Science Publishers, 2010.

[12] D. Chen, S. Chilstedt, C. Dong, and E. Pop, "What Everyone Needs to Know about Carbon-Based Nanocircuits," Online Knowledge Center, IEEE/ACM Design Automation Conference, 2010.

[13] H.-S.P. Wong et al., "Carbon Nanotube Electronics – Materials, Devices, Circuits, Design, Modeling, and Performance Projection," *IEEE Intl. Electron Devices Meeting*, Washington D.C., Dec. 2011.

[14] G. Fiori and G. Iannaccone, "Simulation of Graphene Nanoribbon Field-Effect Transistors," *IEEE Trans. Electron Devices*, vol. 28 no. 8, pp. 760-762, July 2007.

[15] Y. Yoon et al., "Performance Comparison of Graphene Nanoribbon FETs with Schottky Contacts and Doped Reservoirs," *IEEE Trans. Electron Devices*, vol. 55 no. 9, pp. 2314-2323, Sept. 2008.

[16] Y. Ouyang et al, "Comparison of performance limits for carbon nanoribbon and carbon nanotube transistors," *APL*, vol. 89 no. 20, pp. 203107–203109, Nov. 2006.

[17] G. Liang et al., "Performance Projections for Ballistic Graphene Nanoribbon Field-Effect Transistors," IEEE *Trans. Electron Devices*, vol. 54, no. 4, pp. 677–682, Apr. 2007.

[18] P. Michetti and G. Iannaccone, "Analytical Model of One-Dimensional Carbon-Based Schottky-Barrier Transistors," *IEEE Trans. Electron Devices*, vol. 57 no. 7, pp. 1616 – 1625, July 2010.

[19] M. Choudhury et al., "Technology exploration for graphene nanoribbon FETs," *IEEE/ACM DAC*, pp. 272-277, 2008.

[20] S. Frégonèse, C. Maneux, and T. Zimmer, "A versatile compact model for ballistic 1D transistor: GNRFET and CNTFET comparison," *Solid-State Electronics* vol. 54, pp. 1332–1338, 2010.

[21] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *IEEE/ACM DAC*, 2002, pp. 556 - 561.

[22] A. Agarwal, D. Blaauw, V. Zolotov, and S. Vrudhula, "Computation and refinement of statistical bounds on circuit delay" in *IEEE/ACM DAC*, 2003, pp. 348–353.

[23] L. Jing-Jia, A. Wang, and L.C. Cheng, "False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation," in *IEEE/ACM DAC*, 2002, pp. 566 - 569.

[24] A. Ramalingam et al., "An Accurate Sparse Matrix Based Framework for Statistical Static Timing Analysis," in *IEEE/ACM ICCAD*, 2006, pp. 231 - 236.

[25] H. Chang, V. Zolotov, S. Narayan, and C. Visweswariah, "Parameterized Block-Based Statistical Timing Analysis with Non-Gaussian Parameters, Nonlinear Delay Functions," in *IEEE/ACM DAC*, 2005.

[26] J. Kao, S. Narendra, and Chandrakasan A., "Subthreshold leakage modeling and reduction techniques," in *IEEE/ACM ICCAD*, 2002.

[27] H. Chang and S. Sapatnekar, "Prediction of leakage power under process uncertanties," *ACM TODAES*, vol. 12, no. 2, pp. 1-27, Apr 2007.

[28] J. Jaffari and M. Anis, "Statistical Thermal Profile Considering Process Variations: Analysis and Applications," *IEEE TCAD*, vol. 27, no. 6, June 2008.

[29] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," *IEEE TCAD*, vol. 26, no. 4, pp. 619-631, April 2007.

[30] A. Srivastava, "Accurate and Efficient Gate-Level Parametric Yield Estimation Considering Correlated Variations in Leakage Power and Performance," in *IEEE/ACM DAC*, 2005.

[31] H. Su, "Full chip leakage estimation considering power supply and temperature variations," in *ISLPED*, 2003, pp. 78-83.

[32] C. E. Clark, "The Greatest of a Finite Set of Random Variables," *Operations Research*, vol. 9, pp. 85-91, 1961.

[33] S. K. Chandra, K. Lahiri, A. Raghunathan, and S. Dey, "Considering process variations during system-level power analysis," in *IEEE/ACM ISLPED*, 2006, p. 342–345.

[34] J. Luu et al., "VPR 5.0:FPGA CAD and architecture exploration tools with single-driver routing, heterogeneity and process scaling," in *Intl. Symp. on FPGAs*, 2009.

[35] Y. Yang, Z. Gu, C. Zhu, R. Dick, L. Shang, "ISAC: Integrated Space and Time Adaptive Chip-Package Thermal Analysis," *IEEE TCAD*, vol. 26, no. 1, pp. 86-99, Jan 2007.

[36] X. Yang et al, "Graphene tunneling FET and its applications in low-power circuit design," Great Lakes Symposium on VLSI, 2010.

[37] A. K. Geim, "Graphene: Status and Prospects," *Science*, vol. 324, pp. 1530-1534, 2009.

[38] K. S. Novoselov et al., "Electric field effect in atomically thin carbon films," *Science*, vol. 306, pp. 666–669, 2004.

[39] Y. Zhang et al., "Experimental observation of the quantum Hall effect and Berry's phase in graphene," *Nature*, vol. 438, pp. 201–204, 2005.

[40] C. Berger et al., "Electronic confinement and coherence in patterned epitaxial graphene," *Science*, vol. 312, pp. 1191–1196, 2006.

[41] C. Xu, H. Li, K. Banerjee, "Modeling, Analysis and Design of Graphene Nano-Ribbon (GNR) Interconnects," *IEEE Trans Electron Devices*, vol. 56 no. 8, pp. 1567-1578, 2009.

[42] K. Nakada et al. "Edge state in graphene ribbons: Nanometer size effect and edge shape dependence," *Phys. Rev. B*, Condens. Matter, vol. 54 no. 24, pp. 17954–17961, Dec. 1996.

[43] V. Barone, O. Hod, and G. E. Scuseria, "Electronic structure and stability of semiconducting graphene nanoribbons," *Nano Letters*, vol 6 no. 12, pp. 2748–2754, Dec. 2006.

[44] Y. W. Son, M. L. Cohen, and S. G. Louie, "Energy gaps in graphene nanoribbons," *PRL*, vol. 97 no. 21, pp. 216 803, Nov. 2006.

[45] X. Li, X. Wang, L. Zhang, S. Lee, H. Dai. "Chemically Derived, Ultrasmooth Graphene Nanoribbon Semiconductors," *Science* 319, pp. 1229-1232, February 2008.

[46] X. Wang and H. Dai. "Etching and narrowing of graphene from the edges," *Nature Chemistry*, 2, pp. 661-665, 2010.

[47] K. Nabors and J. White, "FastCap: a multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 10, pp 1447-1459.

[48] D. Jiménez et al., "Unified compact model for the ballistic quantum wire and quantum well metal–oxide–semiconductor field-effect-transistor," *J. Appl. Phys.*, vol. 94, no. 2, pp. 1061–1068, July 2003.

[49] J. Deng and H.-S. P. Wong, "A Compact SPICE Model for Carbon-Nanotube Field-Effect Transistors Including Nonidealities and Its Application - Part I: Model of the Intrinsic Channel Region," *IEEE Trans. Electron Devices*, vol. 54, pp. 3186-3194, 2007.

[50] K. Nagashio, et al, "Metal/graphene contact as a performance killer of ultra-high mobility graphene analysis of intrinsic mobility and contact resistance," *IEEE IEDM*, Dec 2009.

[51] Predictive Technology Model. (2011, Dec.) [Online]. http://ptm.asu.edu/